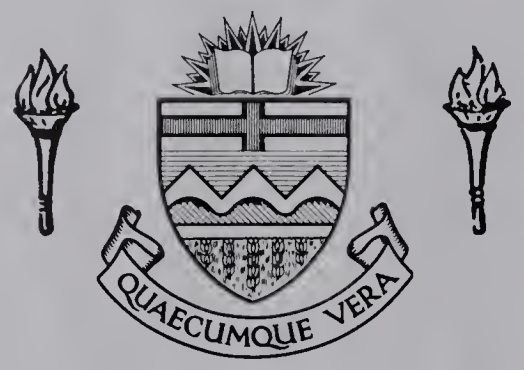


The University of Alberta
Printing Department
Edmonton, Alberta

THE UNIVERSITY OF ALBERTA

A STUDY OF PATTERN CLASSIFICATION

by

© James Wesley Enarson

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL ENGINEERING

EDMONTON, ALBERTA

SPRING, 1969

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled A Study of Pattern Classification submitted by James W. Enarson in partial fulfilment of the requirements for the degree of Master of Science.

ACKNOWLEDGEMENTS

The author would like to express his appreciation to Dr. N. Ahmed for his ideas and discussions on the topics in this thesis and to Dr. V. Gourishankar for his encouragement and guidance in preparing this work. The author would also like to thank his wife for her patience in typing the thesis.

The work was carried out at the Department of Electrical Engineering at the University of Alberta. The financial assistance provided by this Department and the National Research Council is also gratefully acknowledged.

ABSTRACT

The work in pattern recognition and classification is still in its formative stages. The types of classifiers existing have been discussed but many of their properties and capabilities have not been investigated. The training of these machines also requires more study to be fully understood. In this thesis, a review of many of the ideas in pattern classification has been given. Some of these ideas have been extended and enlarged upon.

A recent nonparametric training method has also been investigated. The usefulness of this method is questioned in view of the results of a simulation study discussed in this work. This method is also compared to another recent nonparametric training method.

TABLE OF CONTENTS

	Page
LIST OF FIGURES AND TABLES	(vii)
NOMENCLATURE	(viii)
CHAPTER 1 INTRODUCTION	1
1.1 Some Basic Ideas in Pattern Classification	1
1.2 Scope of the Thesis	2
CHAPTER 2 TYPES OF PATTERN CLASSIFIERS	4
2.1 Introduction	4
2.2 Linear Classifiers	5
2.3 Quadric Classifiers	12
2.4 Φ Classifiers	14
2.5 Layered Machines	18
CHAPTER 3 TRAINING OF PATTERN CLASSIFIERS	24
3.1 Introduction	24
3.2 Parametric Training Methods	25
3.3 Nonparametric Training Methods	31
CHAPTER 4 COMPUTER SIMULATION OF SOME TRAINING METHODS	37
4.1 Introduction	37
4.2 The Classification Problem	38
4.3 Mean-Square-Error Classifier	42
4.4 Probabilistic-Descent Classifier	51
4.5 Comparison of the Two Systems	63
CHAPTER 5 CONCLUSIONS	67
5.1 Summary	67
5.2 Some General Comments	67
5.3 Possible Areas for Further Research	68

Page

BIBLIOGRAPHY 70

APPENDIX A Calculation of the Error Probability for the Baye's Solution 72

APPENDIX B Calculation of the Gradient of the Mean-Square-Error 77

APPENDIX C Calculation of the Error Probability for the Mean-Square-Error Solution 81

LIST OF FIGURES AND TABLES

FIGURES

		Page
2.1	A Linear Machine	6
2.2	A Piecewise-Linear Machine	9
2.3	An Example of Minimum Distance Classification	11
2.4	A Φ Machine	16
2.5	A Layered Machine	19
4.1	Mean-Square-Error Pattern Classifier	46
4.2	Mean-Square-Error and Number of Errors	48
4.3	Decision Boundary for Mean-Square-Error Training Procedure	50
4.4	Number of Errors for Probabilistic-Descent Classifier	57
4.5	Probabilistic-Descent Pattern Classifier	59
4.6	Decision Boundary for Probabilistic-Descent Classifier	61

TABLES

4.1	Summary of Patterson and Womack's Results Together with the Probability of Error From the Exact Baye's Rule Solution	40
4.2	Summary of Simulation Results	66

NOMENCLATURE

Symbol	Meaning
n	dimension of the pattern space
E^n	Euclidean n-dimensional pattern space
A_i	i'th category of the pattern space
R	total number of categories in the pattern space
X	pattern vector
$f_i(X)$	i'th discriminant function
Θ	threshold value for classification
W	weight vector
N	number of points in a finite deterministic pattern space.
$F(X)$	vector output of Φ processor
m	dimension of the vector $F(X)$
I^m	m-dimensional hypercube
$C(\alpha/\beta)$	cost of misclassifying a pattern from the class β by a decision which places it in class α
q_i	probability of pattern class i
$p(X)$	probability of pattern vector X
$p(j/X)$	conditional probability of A_j given X
Σ	variance-covariance matrix of a Gaussian probability distribution
M	mean vector of a Gaussian distribution
E^{n+1*}	augmented weight space
V	solution region of the weight space
V_1	positive half space of a weight space containing two categories
V_2	negative half space of this same weight space

CHAPTER 1 INTRODUCTION

1.1 Some Basic Ideas in Pattern Classification

The process of learning as experienced by humans is not fully understood at the present time. It is known, however, that the ability to learn is very closely associated with the ability to classify data. For example if a child is to learn not to touch hot things it must classify 'things which it finds to be hot' in the category 'things it shouldn't touch'. The data in this case is the description of the hot item. The child may touch this item more than once before making the correct mental classification. There exists a certain training period common to all learning. The length of the training period will be directly related to the extent of the consequences of misclassification, namely, the burning sensation which accompanies touching the hot item. All these facts plus many more which are not fully understood are integrated in some complex way to produce learning in the human mind.

The idea of building a machine which is able to learn has been studied for the past fifteen or twenty years. One method of approach to this problem has been portrayed in the research into pattern recognizers and classifiers. The physical phenomena, of which one was mentioned above, has been studied and an effort to mechanically or electronically classify data has been made. Widrow[1,2] and Huber[3] have published papers describing machines which have been successfully tested on classification problems. The success of these attempts has resulted in useful applications of such machines[4,5].

To formulate the problem mathematically we must first define the physical phenomena using mathematical terminology. Before classification can begin, a set of measurements of the significant properties

of the object or phenomena to be classified must be obtained. (Deciding what constitutes a significant property is in itself a study towards which a vast amount of research has been directed.) Some specific number, say n, of independent measurements will be made on each item to be classified. These n measurements will be used as the components of an n-dimensional vector called the pattern vector. In this way each item considered can be uniquely associated with a point in a Euclidean n-dimensional space, E^n. This space, known as the pattern or measurement space, is then divided into appropriate sets for classification by the use of sets of functions known as discriminant or decision functions. Equating these discriminant functions to zero yields a set of hypersurfaces in the pattern space known as decision surfaces. The positioning of these surfaces is determined by the discriminant function which is a function of the pattern vector together with a set of weighting variables. These weighting variables form a vector called the weight vector. The training process for the machine consists of moving the decision surfaces in E^n in an effort to recognize or classify patterns. This is done by adjusting the weight vectors.

As was stated previously, there exist consequences for misclassification which in many cases will vary with the category from which the pattern arises. This phenomenon is called the loss or cost function, since the consequences of misclassification can be considered as a loss incurred by wrongly classifying a pattern of one category as one belonging to another category.

1.2 Scope of the Thesis

The aim of this thesis will be two-fold. First a general

description of several types of learning machines and some theorems relating to their existence and use will be given. This will be followed by a discussion of training methods for these machines. The reasons for the inclusion of this part are to give a brief view of the present state-of-the-art in this area and also to verify and extend some of the available theorems. The second part of the thesis will include a particular pattern classification problem taken from a recent paper. The methods used in this paper to attack the problem will be stated and analyzed, and a comparison of these methods will be given. It is hoped that this discussion will provide some insight into the problems that have to be faced by researchers in the area of pattern recognition and classification.

Chapter two includes a discussion of various types of classifiers. The first four types of machines mentioned receive only brief explanations. The reader who is interested in these types of classifiers can find very comprehensive discussions on them in the references and also in the many other papers available on the subject. The section on layered machines, however, is a little more comprehensive. This is because the current literature has not, in general, included this type of classifier. The results presented in section 2.5 are for the most part original and it is felt that they show the usefulness of such a classifier.

Chapter three attempts to give a general view of both the parametric and the nonparametric training methods. Chapter four gives the results of a simulation study of a classification problem using two of these algorithms.

CHAPTER 2 TYPES OF PATTERN CLASSIFIERS

2.1 Introduction

The qualitative description of pattern classification can be translated into mathematical terms by using set theoretic concepts. Let $A_1, A_2, \ldots A_R$ denote the sets of measurements or properties relating to R classes or categories of patterns in the n-dimensional Euclidean space from which the pattern vectors are taken. Then the pattern space, E^n, can be considered to be the union of the pattern classes. That is:

$$E^n = A_1 \cup A_2 \cup A_3 \cup \ldots \cup A_R \tag{2.1}$$

The basic requirement for pattern recognition is that the sets $A_1, A_2, \ldots, A_R$ must be non-overlapping or their intersection must be an empty or null set. That is:

$$A_1 \cap A_2 \cap A_3 \cap \ldots \cap A_R = \emptyset \tag{2.2}$$

Equation 2.2 may be valid either in the probabilistic or deterministic sense. By probabilistic sense we mean $A_i \cap A_j = \emptyset$ with a probability approaching one but not necessarily equal to one.

The next step is to attempt to separate these pattern classes. This can be done by a set of functions $f_i(X)$; i = 1,2, ..., R; known as discriminant functions and having the property

$$f_i(X) > f_j(X) \quad \text{for } X \in A_i \quad (i,j = 1,2, \ldots, R; \; i \neq j) \tag{2.3}$$

These functions can take on one of a number of forms; linear, piecewise linear, polynomial, etc. The types of classifiers are named after the types of discriminant functions which they simulate. The various types of classifiers will be discussed in the sections that follow.

2.2 Linear Classifiers

Of all classifiers the linear classifier is the one which has been analyzed most thoroughly. A linear classifier can be described as an implementation of a set of n-dimensional hyperplanes. The mathematical analysis of a linear classifier is by far the most straight forward and the implementation of such a machine can be achieved using very inexpensive but reliable networks.

Such a machine is depicted in Figure 2.1 where the necessary components are a set of weighting and summing devices. A resistive adder network would be sufficient for such a job.

Consider a two-dimensional pattern space containing two categories, A_1 and A_2. A linear classifier can be made which assigns an unclassified pattern X_i to category A_1 if $W^t \cdot X_i > \Theta$ and to A_2 otherwise. The discriminant function in this case is

$$f(X) = W^t \cdot X = w_1 x_1 + w_2 x_2 \tag{2.3}$$

where

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \text{and} \quad W = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

and superscript t denotes transposition. The symbol Θ represents a threshold value for classification. This threshold value is made a part of the discriminant function by many authors by increasing the dimension of the weight vector by one and augmenting the pattern vector with a term equal to one for all patterns, i.e.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} \quad \text{and} \quad W = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

The criterion for classification will then be $X_i \in A_1$ if $f(X_i) > 0$ and $X_i \in A_2$ otherwise, with $f(X_i)$ being defined in terms of these new pattern

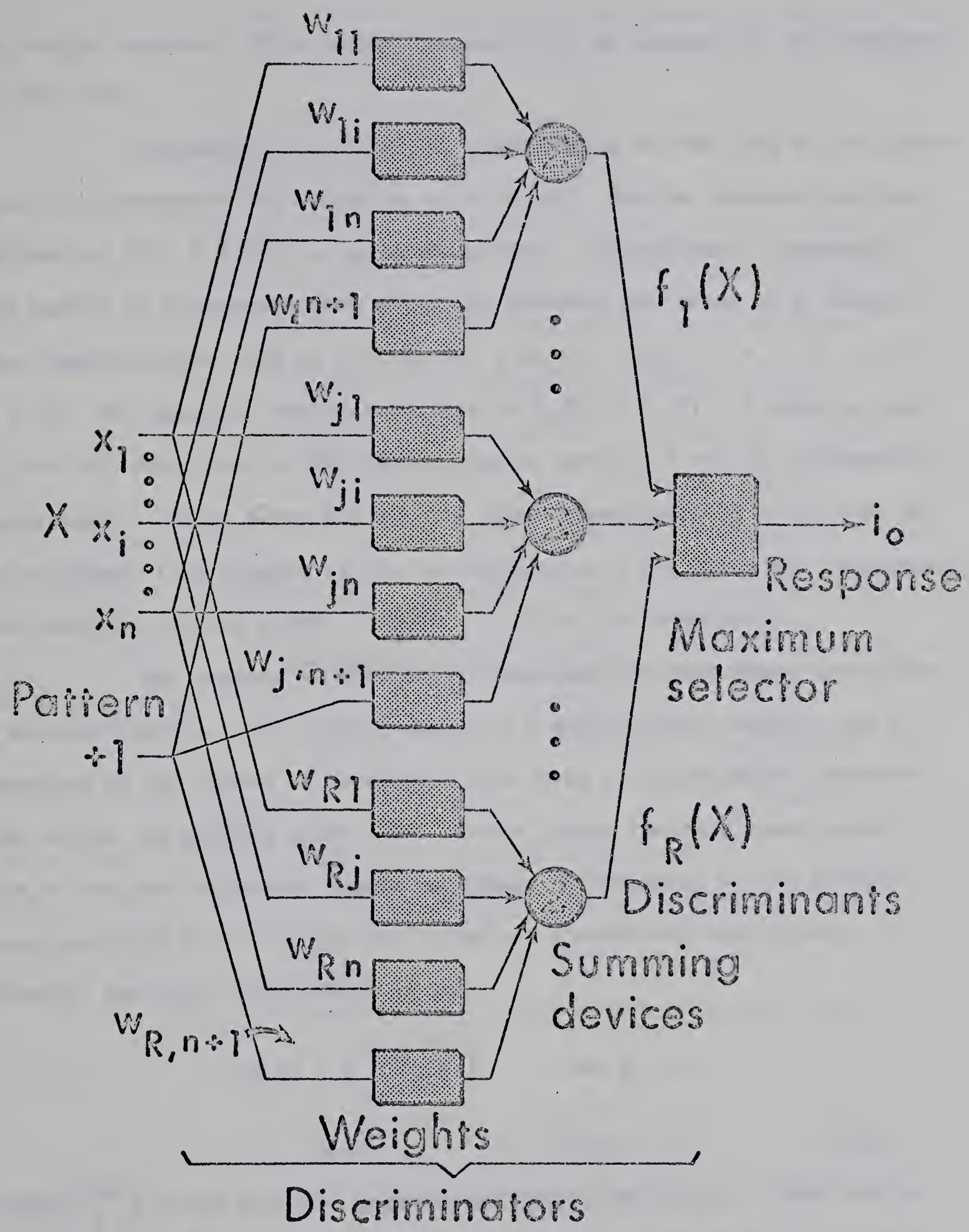


Fig. 2.1 A linear machine

and weight vectors. This latter approach will be assumed in the remainder of this work.

Increasing the number of measurements on the item to be classified to n increases the dimension of X to n+1. The new decision surface defined by f(X) = 0 will be an n-dimensional hyperplane. Increasing the number of categories from which the patterns may arise to R changes the classification rule to $X_k \in A_i$ for $f_i(X_k) > f_j(X_k)$ (j = 1, 2, ..., R; $j \neq i$). The decision surfaces defined by $f_i(X) - f_j(X) = 0$ where A_i and A_j are adjoining sets in the pattern space, make up a set of n-dimensional hyperplanes. There exist R(R-1)/2 of these hyperplanes of which some may be redundant (for example if the two categories a and b are not contiguous the decision surface given by $f_a(X) - f_b(X) = 0$ is redundant).

Now suppose we have an n-dimensional pattern space containing N distinct points. The effectiveness of a discriminant function can be measured by the number of categories this type of discriminant function can divide the pattern space into. For a linear function there exist L(N,n) distinct divisions (known as linear dichotomies) of the pattern space provided no n+1 points lie on an n-1 dimensional hyperplane. Nilsson[7] has shown this number to be

$$L(N,n) = 2\sum_{i=0}^{n}\binom{N-1}{i} \qquad \text{for } N > n$$

$$= 2^N \qquad \text{for } N < n \qquad (2.4)$$

where $\binom{N-1}{i}$ is the binomial coefficient $(N-1)!/(N-1-i)!i!$. This can be compared with 2^N which is the total number of possible dichotomies of N points.

Highleyman[8] states that the optimal decision function for the following two class problems are linear.

1) The two classes are equally probable a priori, have equal

losses associated with misrecognition, and have probability distributions over the measurement space which are unimodal, spherically symmetrical, and identical except for a displacement of modes.

2) The two classes are equally probable a priori, have equal losses associated with misrecognition, and have probability distributions over the measurement space which are Gaussian and which have equal covariance matrices.

3) The convex hulls of the points in measurement space contained in each pattern class are nonintersecting.

These conditions are, however, more restrictive than is necessary. The conditions of equal probability and equal losses in (1) and (2) only serve to vary the threshold Θ and so inequality of either the probabilities or losses or both would still give a linear solution.

One very important application of linear classifiers is the piecewise linear machine. Such a machine can be used in situations where the minimum distance between a pattern and a category is the criterion for classification. This distance is usually defined in terms of a norm in the pattern space.

The discriminant function for a piecewise linear classifier is given by

$$f_i(X) = \max_{j = 1, \ldots, L_i} \{g_i^j(X)\} \quad (i = 1, \ldots, R) \tag{2.5}$$

where $g_i^j(X)$, called a subsidiary discriminant function, is a linear function of X. $(g_i^j(X) = W^t X)$. Such a machine is shown in Figure 2.2. While the discriminant function of this machine is linear for all X the addition of a second maximum selector results in the machine response

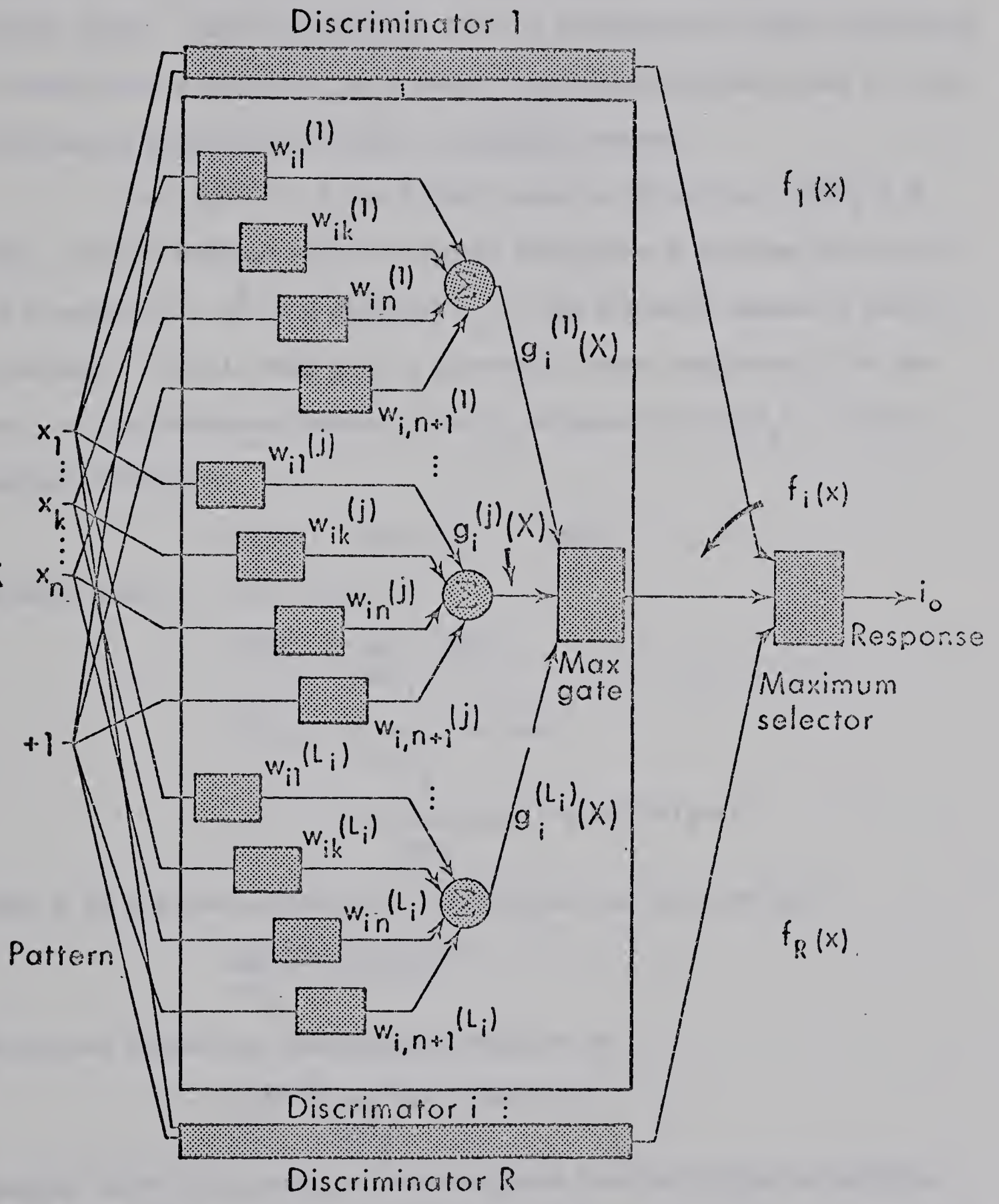


Fig.2.2 A piecewise linear machine

suddenly changing from one subsidiary function to another. The decision surfaces will therefore be made up of portions of hyperplanes in the pattern space. These hyperplanes will be connected but their directions are completely arbitrary. As a result the categories described by these discriminant functions are not, in general, convex.

Let $P_1, \ldots, P_R$ be R point sets in E^n and let $P_i \cap P_j = \emptyset$ $(i \neq j)$. Let us suppose that we wish to synthesize a machine which will put a pattern $X \in E^n$ in a category P_i if the distance between X and P_i is minimum. We will show that a piecewise linear machine will do the job. Let the distance between X and P_i be given by $d(X,P_i)$. We will conclude $X \in P_k$ if

$$d(X,P_k) < d(X,P_j) \qquad (j \neq k)$$

Defining $d(X,Y) = |X-Y|$ gives

$$d(X,P_i) = \inf_{p \in P_i} |X-p|$$

$$d^2(X,P_i) = \inf_{p \in P_i} (X-p,X-p)$$

$$= 2 \sup_{p \in P_i} \{(p \cdot X) - \tfrac{1}{2}(X \cdot X) - \tfrac{1}{2}(p \cdot p)\}$$

Since X is a fixed pattern, it is sufficient to consider only

$$\sup_{p \in P_i} \{(p \cdot X) - \tfrac{1}{2}(p \cdot p)\}$$

We can now define our discriminant function as

$$f_i(X) \triangleq \sup_{p \in P_i} \{(p \cdot x) - \tfrac{1}{2}(p \cdot p)\}$$

However, if P_i is a point set of continuum this will give an infinite (nondenumerable) family of discriminant functions and thus will not be of any practical use. But, since $P_i \cap P_j = \emptyset$ for all i, a countable set of points P_{i_k} can be chosen to represent the category i, such that

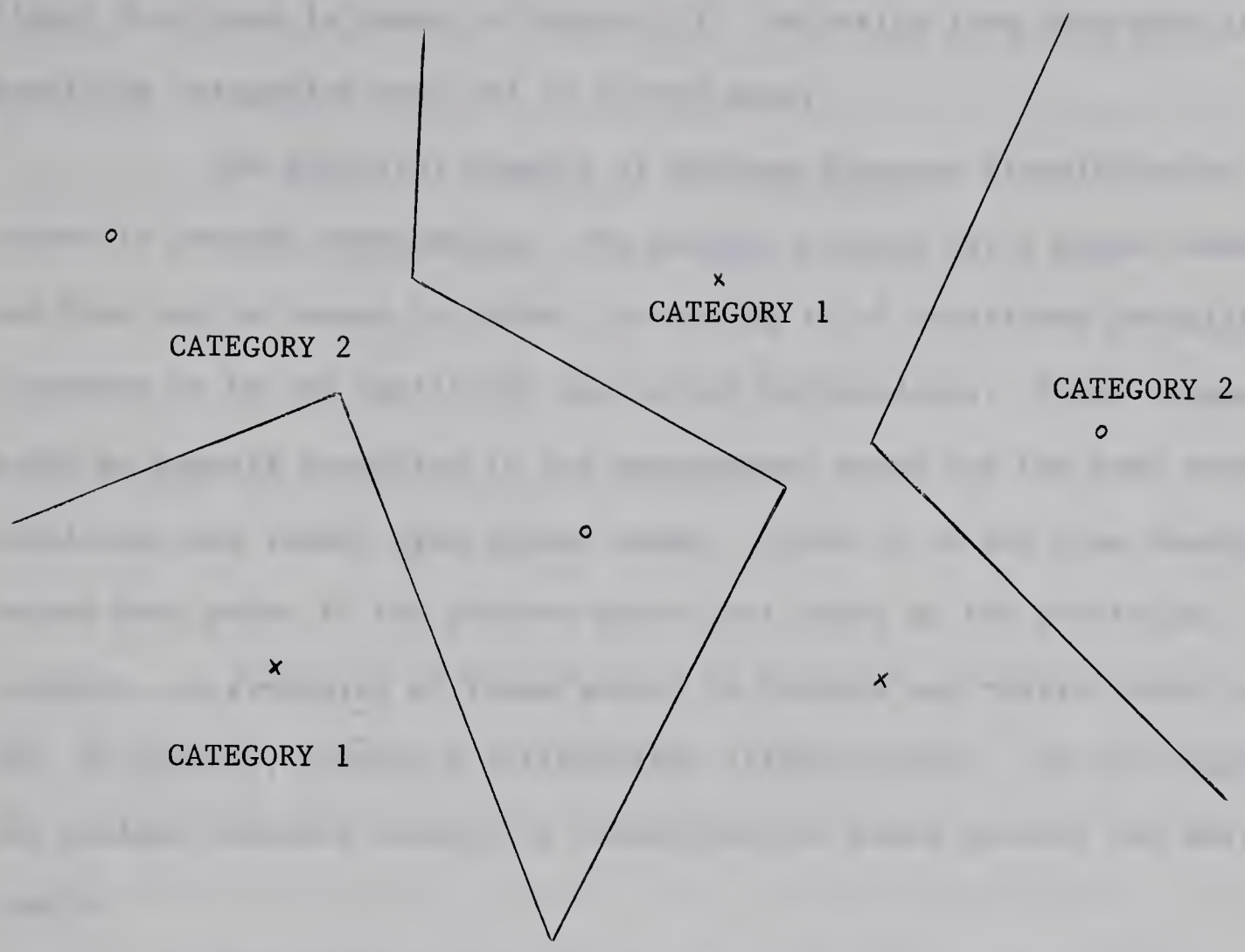


FIGURE 2.3 AN EXAMPLE OF MINIMUM DISTANCE CLASSIFICATION

$$\bigcup_{k=1}^{L_i} N_\varepsilon(P_{i_k}) \supset P_i$$

where $N_\varepsilon(P_{i_k})$ is an ε neighborhood of P_{i_k} ε P_i. And so we have

$$f_i(X) = \max_{k = 1, \ldots, L_i} g_{i_k}(X, P_{i_k}) \qquad (2.6)$$

where

$$g_{i_k}(X, P_{i_k}) = (P_{i_k} \cdot X) - \tfrac{1}{2}(P_{i_k} \cdot P_{i_k})$$

g_{i_k} is a linear function of X and W (the P_{i_k} will form the components w_i of W). This is of the same form as equation 2.5 which is the discriminant function of a piecewise linear machine. An example of a two-dimensional pattern space divided into two categories by piecewise

linear functions is shown in Figure 2.3. We notice from here that the resulting categories need not be convex sets.

One practical example of minimum distance classification arises in weather forecasting. The weather pattern for a given location and time may be caused by either the moving in of conditions prevailing elsewhere or by the particular geological surroundings. These causes would be greatly separated in the measurement space but the same weather conditions may result from either cause. Since it is the area immediately around each point in the pattern space that makes up the particular category, an averaging of these points to produce one central point would not, in general, produce a satisfactory categorization. In such cases, the minimum distance concept of classification would produce the best result.

The minimum distance classifier is a very effective classifier if the position of the categories in the pattern space is known a priori. However, if this knowledge is unavailable the classification problem becomes very difficult due to the difficulties of training a piecewise linear machine. Such a training procedure requires not only an adjustment of the weight values but also an adjustment in the number of subsidiary discriminant functions in each discriminator. As yet no one has come up with such a training method and so this problem would pose a very interesting research project for anyone interested.

2.3 Quadric Classifiers

The linear machines just discussed were a special case of polynomial machines whose study comes under the heading Φ machines. However, one special case of polynomial machines, namely, the quadric

machine, can produce an optimal solution for an important class of problems. For this reason we shall take a look at it separately and this will also give an introduction to Φ machines.

The quadric discriminant function takes the form

$$f_i(X) = \sum_{j=1}^{n} w_{jj} x_j^2 + \sum_{j=1}^{n-1} \sum_{k=j+1}^{n} w_{jk} x_j x_k + \sum_{j=1}^{n} w_j x_j + w_{n+1} \tag{2.7}$$

which can be written in matrix notation as

$$f_i(X) = X^t A_i X + X^t B_i + C_i \tag{2.8}$$

The decision surfaces obtainable with a quadric machine are sections of second-degree surfaces. These include hyperellipsoids for A_i positive definite which form hyperspheres if A_i is a diagonal matrix, hyperellipsoidal cylinders for A_i positive semidefinite and hyperhyperboloids for other types of A_i.

If equation 2.7 were written out in full it would take the form

$$f_i(X) = w_{11}x_1^2 + w_{22}x_2^2 + \ldots + w_{12}x_1x_2 + w_{23}x_2x_3 + \ldots + w_1x_1 + w_2x_2 + \ldots + w_{n+1} \tag{2.9}$$

Suppose a mapping F exists such that F: $E^n \rightarrow E^m$ (m>n) in such a way that

$$F(X) = (x_1^2, x_2^2, \ldots, x_1x_2, x_2x_3, \ldots, x_1x_2, \ldots, 1) \tag{2.10}$$

We can now write

$$f_i(X) = W_i^t F(X) \tag{2.11}$$

where

$$W_i = (w_{11}^i, w_{22}^i, \ldots, w_{12}^i, w_{23}^i, \ldots, w_1^i, w_2^i, \ldots, w_{n+1}^i)$$

W and F(X) will contain

$$m = n + n(n-1)/2 + n+1$$
$$= (n^2 + 3n + 2)/2 \qquad (2.12)$$

terms where n is the dimension of the pattern vector. If a processor were to be constructed whose output was the vector F(X) for an input of the pattern vector X, the quadric classifier could be built by connecting the output of such a processor to the input of an m-dimensional linear machine. The processor would consist of a set of multipliers which, although not as economical or reliable as the network making up the linear classifier, are readily available.

A quadric classifier is able to produce the optimal solution for a two category problem in which the patterns within the categories have a Gaussian distribution. Problem (2) on page 8 is a special case of this general problem. The actual derivation of this discriminant function will be given in Chapter 3. Since many of the natural observations which one would wish to categorize can be approximated by a Gaussian distribution the quadric machine is a very important one.

2.4 Φ Classifiers

As we have seen in the last section the quadric machine can be built by first building a processor to produce F(X) and then using an m-dimensional linear machine for classification. Extending this idea, suppose $F: E^n \rightarrow E^m$ $(m>n)$ such that

$$F(X) = \{F_1(X), F_2(X), \ldots, F_m(X)\} \qquad (2.13)$$

where $F_i(X)$ are polynomial functions of X. A processor can be built to produce such an F(X) from the pattern vector X and its output can be connected to the input of a linear machine. The resulting discriminant function is a linear function of the weight vector elements, w_i, and can be written in the form

$$\Phi(X) = w_1F_1(X) + w_2F_2(X) + \ldots + w_mF_m(X) \qquad (2.14)$$

Such a machine is called a Φ machine and is shown in Figure 2.4. If F is the identity operator, F(X) = X, the resulting machine is linear. If F maps X into F(X) of the form $x_i x_j$ $(i,j = 0, 1, \ldots, n)$ $(x_0 = 1)$ the resulting machine is quadric. If F maps X into F(X) of the form

$$x_{k_1} x_{k_2} x_{k_3} \ldots x_{k_r} \quad (k_1, k_2, \ldots, k_r = 0, 1, 2, \ldots, n)$$

the resulting machine is an r'th order polynomial machine. The processor in all cases consists of a set of multipliers.

These polynomial machines will require a processor and need more weighting devices than the linear machine and, therefore, will be less economical. The main justification for their development and use lies in their increased ability to classify. From equation 2.4 we know the number of linear dichotomies of N points is

$$L(N,n) = 2\sum_{i=0}^{n}\binom{N-1}{i} \quad \text{for } N>n$$
$$= 2^N \quad \text{for } N\leq n$$

In the case of Φ machines we have a linear function of the weight vector in a higher dimensional space. This means we are attempting to separate the mapped categories in E^m by linear functions. But equation 2.4 gives the number of linear dichotomies of N points in a space of specified dimensions. Therefore, it follows that the number, say P, of distinct

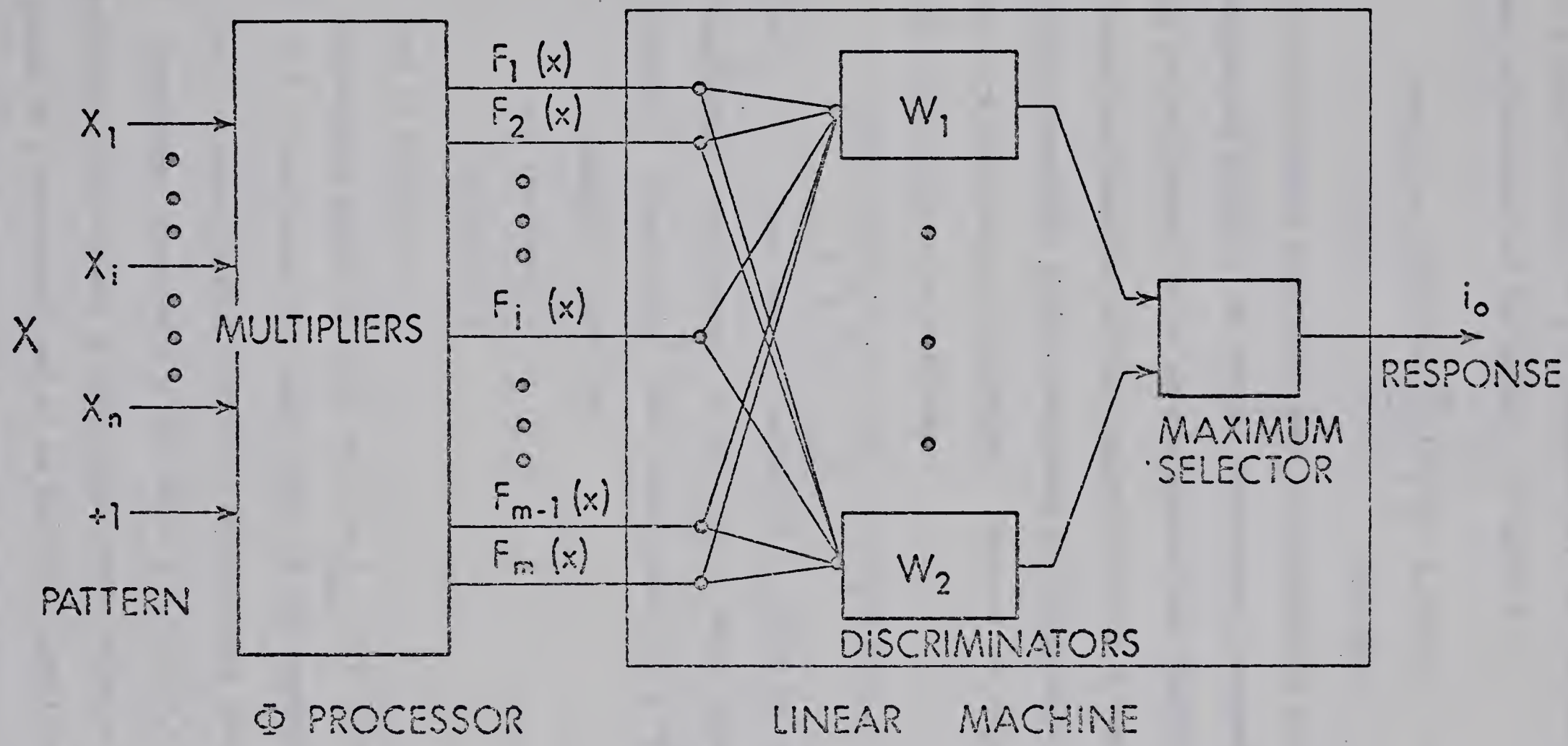


Fig. 2.4 A Φ MACHINE

divisions of the pattern space by a Φ function will be the same as the number of linear dichotomies of the mapped, E^n space i.e.

$$P(N,n) = L(N,m) = 2\sum_{i=0}^{m}\binom{N-1}{i} \quad \text{for } N>m$$
$$= 2^N \quad \text{for } N\leq m \qquad (2.15)$$

where m>n. So the increased complexity may be justified by the increased classification ability. As before this result is based on the assumption that no n+1 points lie on an n-1 dimensional hyperplane.

Since the dimension of this mapped space is a function of the degree of the polynomials $F_i(X)$, the number of dichotomies of this space will also be a function of this degree. Suppose the degree of each $F_i(X)$ is r, i.e. $F_i(X)$ is of the form

$$F_i(X) = x_{k_1} x_{k_2} x_{k_3} \ldots x_{k_r} \qquad k_i = 0, 1, 2, \ldots n$$

where we define $x_0 = 1$. There are n+1 variables x_j which we wish to group in distinguishable arrangements of the size r, each arrangement forming one dimension. From Feller[6] this would result in $m = \binom{n+r}{r}$. In the case of r = 2 (quadric classifier)

$$m = (n+2)(n+1)/2$$
$$= (n^2+3n+2)/2$$

which is the same as equation 2.12.

The maximum degree of $F_i(X)$ will be $r_{max} = n$ which gives $m_{max} = (2n)!/(n!)^2$. This means the upper limit of the summation in equation 2.15 will be much greater than the corresponding limit in equation 2.4. Therefore, provided N>m the ratio P(N,n)/L(N,n) will be large. In practical applications N often approaches infinity (i.e. the pattern space is a continuum) and so we see the classification ability of the polynomial machine will greatly exceed that of a linear

machine.

2.5 Layered Machines

As an extension of the ideas presented in the last section, one could consider the possibility of producing a classifier by simply interconnecting a number of linear classifiers in any one of a number of ways. The output from a group of linear machines can be used as the input to another linear machine. The analysis of such an interconnected network of linear machines is, in general, very difficult since no one has yet come up with a transfer function representation of a linear machine.

If we restrict the interconnection of these linear machines such that only connections between layers of linear machines are allowed with no connections between separate machines within these layers occurring, we can derive some interesting results. Under such restrictions the outputs of one layer of linear classifiers are used as an input to the next layer. These machines are known as layered machines, an example of which is shown in Figure 2.5. To simplify our analysis of such a machine it will be assumed in the work that follows that the pattern space contains only two categories which are to be dichotomized. This means the +1 or -1 output of the final layer is the response of the entire machine.

From the diagram we see that the input to the first layer is the pattern vector while the input to each subsequent layer is the output of the previous layer. This output, which can be considered a vector, contains elements restricted to take on the values of +1 and -1. If we assume that there exist d_1 linear machines in the first layer and

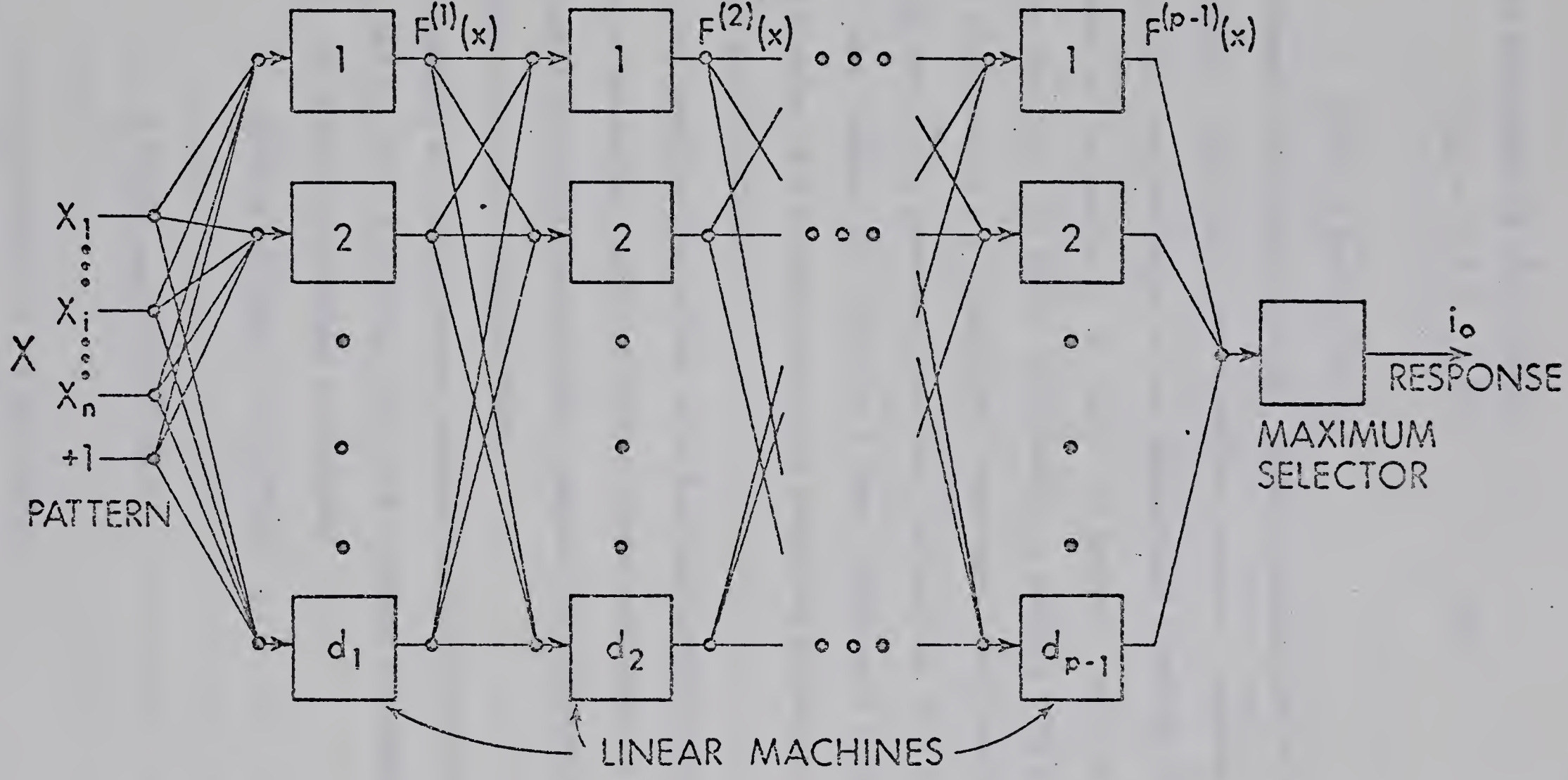


Fig. 2.5 A LAYERED MACHINE

the output of this layer is $F^1(X)$, then

$$F^1(X) = \{F_1{}^1(X), F_2{}^1(X), \ldots, F_{d_1}{}^1(X)\}$$

where

$$F_i{}^1(X) = \text{sgn}\ (W_i{}^1 . X)$$

and sgn represents the signum function. Similar expressions exist for the other layers. This first layer therefore produces a mapping of the pattern space into the verticies of a d_1-dimensional hypercube with the origin situated in the centre of the cube. In general, the j'th layer with an input from the i'th layer, will produce a mapping from a d_i-dimensional hypercube to a d_j-dimensional hypercube. The final layer maps the response of the previous layer into the vertices of a one-dimensional cube, namely, two points on a line. Since each element of the layered machine is a linear machine the resulting divisions of the pattern space must be linear.

The question then arises as to how many layers are required to dichotomize the pattern space or whether there even exists a finite number of layers which will successfully complete the dichotomization? Let us consider the second question first.

Suppose the pattern space contains two nonintersecting subsets, A_1 and A_2; $E^n = A_1 \cup A_2$ and $A_1 \cap A_2 = \emptyset$. Define a d-dimensional hypercube by I^d, $I^d \subset E^d$. As before we consider

$$F^p(X) = \{F_1{}^p(X), \ldots, F_{d_p}{}^p(X)\}$$

where

$$F_i{}^p(X) = \text{sgn}\ (W_i{}^p . F^{p-1}(X))$$

Let

$$Z_1 = \{z \in I^d : z = F_p(X), X \in A_1\};$$

$$Z_2 = \{z \varepsilon I^d : z = F_p(X),\ X \varepsilon A_2\}$$

We then have the following theorem:

Theorem 2.1: If two categories A_1 and A_2 exist in the pattern space such that $A_1 \cap A_2 = \emptyset$ and there exists some mapping B: $E^n \rightarrow I^d$ then there exists a finite number p of layers in a layered machine which will produce a linear dichotomy of Z_1 and Z_2 i.e. $CoZ_1 \cap CoZ_2 = \emptyset$, where Z_1 and Z_2 are as previously defined, and "Co" means convex hull.

Proof: We want to know if there exists $p<\infty$ such that $CoZ_1 \cap CoZ_2 = \emptyset$ when $A_1 \cap A_2 = \emptyset$. Suppose there does not exist any such finite p. Then $\lim_{p\rightarrow\infty} (CoZ_1 \cap CoZ_2) \neq \emptyset$. This means there exists a Z_0 for all $p<\infty$ such that $Z_0 \varepsilon (CoZ_1 \cap CoZ_2)$, which implies $Z_0 \varepsilon CoZ_1$, $Z_0 \varepsilon CoZ_2$ for all p. But because each Z in Z_1 and Z_2 is a binary vector, the points making up the sets are discrete so that $Z_0 \varepsilon CoZ_1$ implies $Z_0 \varepsilon Z_1$ and $Z_0 \varepsilon CoZ_2$ implies $Z_0 \varepsilon Z_2$. Since each F(X) is a single valued function there must exist an X_0 such that $F_p(X_0) = Z_0$ for $X_0 \varepsilon A_1$ and $F_p(X_0) = Z_0$ for $X_0 \varepsilon A_2$; which implies $X_0 \varepsilon A_1 \cap A_2$, meaning $A_1 \cap A_2 \neq \emptyset$. But this contradicts the original premise that $A_1 \cap A_2 = \emptyset$. Therefore, there must exist a $p<\infty$. Q.E.D.

We now state a theorem which we can use to establish how many layers are needed to dichotomize a pattern space.

Theorem 2.2: If there exist m hyperplanes which form a nonredundant partition of the two finite distinct point sets A_1 and A_2, a sufficient condition that $CoZ_1 \cap CoZ_2 = \emptyset$ is that exactly m+1 cells formed by the partition be occupied by elements of A_1 and A_2, where

$$Z_1 = \{z \varepsilon I^m : z = F(X),\ X \varepsilon A_1\} \text{ and } Z_2 = \{z \varepsilon I^m : z = F(X),\ X \varepsilon A_2\}$$

and F(X) and I^m are as defined in Theorem (2.1). These point sets can then be separated by a two-layer machine.

The term nonredundant partition here means a partition with the property that if one of the separating hyperplanes is removed, at least two nonempty cells will merge into one cell. A proof of this theorem can be found in Nilsson[7].

If the d partitioning hyperplanes of the pattern space were all parallel it would follow that there were exactly d+1 nonempty cells. If we restricted the subsets of the pattern space to a finite number of distinct points we could further say that these two subsets can be non-redundantly partitioned by a set of parallel hyperplanes. This follows from the fact that there exist only a finite number of lines which join all these points but there exist an infinite number of directions in E^n. To partition these points by a set of parallel hyperplanes we need only choose a direction for these hyperplanes which is different from any of the directions of the lines joining all these points. It can therefore be said that a pattern space containing a distinct finite number of pattern vectors contained in two subsets can be dichotomized by a two-layer machine.

We have already restricted our consideration to pattern spaces for which the mapping B: $E^n \rightarrow I^d$ exists. But, provided $d<\infty$, the verticies of the hypercube I^d form two subsets Y_1 and Y_2 made up of a distinct finite number of points where Y_1 corresponds to the mapping of elements from A_1 and Y_2 to the mapping of elements from A_2. From Theorem (2.2) we can therefore say that a two layer machine will dichotomize the subsets Y_1 and Y_2 in I^d. We have therefore proved the theorem:

Theorem 2.3: Any pattern space containing two subsets for which we can find B: $E^n \rightarrow I^d$ ($d<\infty$) can be dichotomized by a three layer machine, the first layer being the mapping B.

So far we have assumed that each element of the layered machine is a linear classifier. In such a case, for the mapping B to exist there must exist a set of hyperplanes which divide the pattern space into subsets each containing only members from one category.

As has already been implied in the work in this section, the discriminant function for a layered machine employing only linear classifiers is piecewise linear. A derivation of this discriminant function although important is not particularly enlightening and so is not included. Such a derivation can be found in Nilsson[7].

There seems to be no reason for restricting the elements of a layered machine to linear classifiers. In particular if the first layer were a set of Φ machines, a much more complex discriminant function would result and B could be made to map any bounded boundary. To the best of the author's knowledge this problem has never been investigated, although it would appear to have great possibilities. Another problem with layered machines which has not been solved is the problem of training. The availability of several layers, each of which contains several elements, naturally increases the complexity of any training method. There will be a greatly increased number of weights to adjust as well as an increased number of ways of adjusting them. For anyone interested, the work of layered machines offers a number of possible avenues of research.

CHAPTER 3 TRAINING OF PATTERN CLASSIFIERS

3.1 Introduction

In this chapter, we will consider the problem of training a pattern classifier by the use of a set of pattern vectors known as the training sample. This consists of varying the elements of the weight vector W until a stage is reached where any further change in W would increase the number of errors given by this classification procedure.

In section 2.1, it was stated that the pattern space can be separated into appropriate pattern classes by the set of discriminant functions $f_i(X)$ $(i = 1, 2, \ldots, R)$ if this set of functions exist such that $f_i(X) > f_j(X)$ for $X \varepsilon A_i$ $(i, j = 1, 2, \ldots, R; i \neq j)$. In the discussion in Chapter 2 on the various types of classifiers, it was found that the discriminant function $f_i(X)$ is always of the general form $f_i(X) = (W \cdot \Phi(X))$ where $\Phi(X)$ is some function of the pattern vector. It should be noted that throughout this discussion the function $\Phi(X)$ is assumed to take on many different forms varying from a linear to an r'th order polynomial function of X, but $f_i(X)$ is always a linear function of the weight vector W.

There are two general types of training methods, the parametric and the nonparametric methods. In many classification problems, the pattern classes are defined by sets of parameters. For instance, when the patterns are random variables, each class is specified by some distinct probability function. The parametric training method consists of estimating the values of the parameters defining the sets to be classified from a training sample and determining a suitable discriminant function based on these parameters and the values of the loss functions.

However, the type of probability distribution of pattern vectors in the pattern space may not be known. A different type of training, a non-parametric method, is required for this type of problem. In this case, one of several methods can be used to search for an appropriate discriminant function based on a pre-established risk function. The training procedure then uses this search technique to determine a weight vector through an iterative procedure which produces a minimum of the risk function.

3.2 Parametric Training Methods

To begin our analysis of training methods, we must first define our cost function. Let $C(\alpha/\beta)$ (≥ 0) be the cost of misclassifying a pattern from the class β by a decision which places it in class α. $\{C(\alpha/\alpha) = 0\}$. The value of this must be known a priori but in many cases a rough estimate of its relative value is sufficient. The optimum discriminant function will then be defined as the one which minimizes the cost of misclassification.

Suppose, as was done in Chapter 2, that the pattern space, E^n, consists of R categories A_i $(i = 1, \ldots, R)$. That is

$$E^n = A_1 \cup A_2 \cup \ldots \cup A_R \tag{3.1}$$

We can derive a discriminant function which will divide the pattern space into these R categories by using conditional probabilities. Let the probability distribution of pattern vectors in category A_i be given by $p(X/i)$. For the parametric training procedure, we either assume or try to approximate the form of the density function. In some cases it may even be known. To fully establish the statistics of the classification problem, we must determine the probabilities $q_i (i = 1, \ldots, R)$

of each pattern class and the parameters which specify each p(X/i) from the training samples given.

Now suppose we are given a pattern vector X. The probability that it comes from some particular category A_j is p(j/X) i.e. the conditional probability of category A_j given X. The cost of such a classification is C(j/i) where A_i is the category to which X belongs. The average expected cost will then be

$$K(i) = \sum_{j=1}^{R} C(j/i)\ p(j/X) \qquad (3.2)$$

The pattern X will be assumed to belong to the A_i for which K(i) is a minimum. In this way, the average cost of misclassification is minimized.

By use of Bayes rule we can write

$$p(j/X) = \frac{p(X/j)}{p(X)}\ q_j \qquad (3.3)$$

where q_j is the probability of occurrence of category j, p(X) is the probability of the pattern X occurring and p(X/j) is the conditional probability of X given that it belongs to category j. Equation 3.2 can now be rewritten as

$$K(i) = \{1/p(X)\} \sum_{j=1}^{R} C(j/i)\ p(X/j)\ q_j \qquad (3.4)$$

Since p(X) will be common to all K(i) (i = 1, 2, ..., R) it can be removed. We can then define our discriminant function as

$$f_i(X) = \sum_{j=1}^{R} C(j/i)\ p(X/j)\ q_j \qquad (3.5)$$

with classification taking place according to the rule

$$f_i(X) > f_j(X) \quad \text{implies } X \in A_j \quad (i,j = 1, 2, \ldots, R;\ i \neq j)$$

From this we see that, assuming C(j/i) is given for all i and j, all that is needed is an estimate of each p(X/j) and the probabilities q_j. A machine employing such a discriminant function is often called a Baye's Machine. Anderson[9] has shown that a Baye's machine yields the optimal solution in that it minimizes the the cost of misclassification.

Consider the case when the training sample contains N_i pattern vectors from category A_i for each i. Since the training sample is supposed to represent the pattern space sufficiently for learning, our best estimate of q_i is

$$q_i = N_i / \sum_{j=1}^{R} N_j$$

The density function p(X/j), is more difficult to find. Methods do exist, however, for approximating the density and the distribution functions from a given set of independent samples taken from the probability space. One such method is given by Kashyap and Blaydon[10]. Other procedures suitable for classification problems are also available. If, however, the form of the density function is known or can be assumed, the problem of training is greatly simplified since we need only estimate the parameters forming the distribution in order to specify p(X/j).

One of the most common distributions of patterns in the pattern space is the Gaussian or normal distribution. The significant parameters of this distribution are the mean and variance. Let the vector whose elements are the mean values of the n variables of the distribution be given by M and let the matrix containing the variances and covariances be Σ. The normal distribution can then be written in

the form:

$$p(X) = \{1/2\pi|\Sigma|^{\frac{1}{2}}\} \exp\{-\tfrac{1}{2}(X-M)^t\Sigma^{-1}(X-M)\} \qquad (3.6)$$

where $|\Sigma|$ denotes the determinant of Σ. So we have

$$p(X/j) = \{1/2\pi|\Sigma_j|^{\frac{1}{2}}\} \exp\{-\tfrac{1}{2}(X-M_j)^t\Sigma_j^{-1}(X-M_j)\} \qquad (3.7)$$

For the two category case the discriminant functions become

$$f_1(X) = C(1/1)\{q_1/2\pi|\Sigma_1|^{\frac{1}{2}}\} \exp\{-\tfrac{1}{2}(X-M_1)^t\Sigma_1^{-1}(X-M_1)\} +$$

$$+ C(2/1)\{q_2/2\pi|\Sigma_2|^{\frac{1}{2}}\} \exp\{-\tfrac{1}{2}(X-M_2)^t\Sigma_2^{-1}(X-M_2)\}$$

and

$$f_2(X) = C(1/2)\{q_1/2\pi|\Sigma_1|^{\frac{1}{2}}\} \exp\{-\tfrac{1}{2}(X-M_1)^t\Sigma_1^{-1}(X-M_1)\} +$$

$$+C(2/2)\{q_2/2\pi|\Sigma_2|^{\frac{1}{2}}\} \exp\{-\tfrac{1}{2}(X-M_2)^t\Sigma_2^{-1}(X-M_2)\}$$

As was mentioned earlier there is generally no loss connected with correct classification (i.e. $C(i/i) = 0$). Because we are dealing with the two category situation we can reduce the discriminant function to $f(X) = f_2(X) - f_1(X)$ with classification taking place according to $f(X)>0$ implies $X \in A_1$ and $f(X)<0$ implies $X \in A_2$. Because the logarithm is a monotonically increasing function of its argument, the same classification would result if we were to consider $\log f_i(X)$ instead of $f_i(X)$. These simplifications reduce the discriminant function to

$$f(X) = \tfrac{1}{2}\{(X-M_2)^t\Sigma_2^{-1}(X-M_2) - (X-M_1)^t\Sigma_1^{-1}(X-M_1) +$$

$$+ 2 \log C(1/2)/C(2/1) + \log |\Sigma_2|/|\Sigma_1| +$$

$$+ 2 \log q_1/q_2\} \qquad (3.8)$$

As can be seen, this function is a quadratic function of X and is the special case mentioned in section 2.3. To obtain the special case

mentioned in section 2.2 case 2 we set $\Sigma_1 = \Sigma_2 = \Sigma$ and obtain

$$f(X) = X^t\Sigma^{-1}(M_1 - M_2) + \tfrac{1}{2}(M_2{}^t\Sigma^{-1}M_2 - M_1{}^t\Sigma^{-1}M_1) + \\ + \log C(1/2)/C(2/1) + \log q_1/q_2 \qquad (3.9)$$

which is a linear function of X.

The parametric training for the case of a Gaussian distribution consists of estimating the values for Σ_i, M_i and q_i for all i, by use of the training samples. Anderson has shown[9] that if a set of N_i samples is available from a category having a Gaussian distribution of pattern vectors the maximum liklihood estimate of the mean is

$$\langle M_i \rangle = \{1/N_i\} \sum_{\alpha=1}^{N_i} X_\alpha \qquad (3.10)$$

and the maximum liklihood estimate of the variance would be

$$\langle \Sigma_i \rangle = \{1/N_i\} \sum_{\alpha=1}^{N_i} (X_\alpha - \langle M_i \rangle)(X_\alpha - \langle M_i \rangle)^t \qquad (3.11)$$

assuming that the measurements in the pattern space were independent. The values of q_i can be estimated as previously shown. Improvements on these estimates can be made if the knowledge of the distribution is increased. For example, if the measurements are not independent, i.e. the covariance terms in Σ_i are non-zero, Anderson shows an improved method of solving for the matrix Σ_i. If the variance-covariance matrix is known a priori an improved estimate of the mean vector can be obtained as shown by Nilsson[7].

In his work in pattern recognition Cooper[11] proposes a probability distribution given by

$$p_m(X) = \{m\Gamma(n/2)/2\Gamma(n/m)\pi^{\frac{1}{2}n}|\Sigma|^{\frac{1}{2}}\} \exp\{-(X-M)^t\Sigma^{-1}(X-M)\}^{\frac{1}{2}m} \tag{3.12}$$

where n is the number of dimensions of the pattern space and Γ is the gamma function. This reduces to the Gaussian distribution for m = 2. The advantage of using this distribution over the Gaussian distribution can be seen from the work done by Ahmed[12] on this function. Ahmed proves the following result. As $m\to\infty$ $p_m(X)$ converges to

$$\begin{aligned} p(X) &= \Gamma(\tfrac{1}{2}n + 1) \;/\; |\Sigma|^{\frac{1}{2}}\pi^{\frac{1}{2}n} && \text{for all } X \in S \\ &= 0 && \text{for all } X \notin S \end{aligned} \tag{3.13}$$

where

$$S = \{X\epsilon E^n : \sqrt{(X-M)^t\Sigma^{-1}(X-M)} \quad < 1\}$$

From this it can be seen that by increasing m the distribution can be made to drop off to zero much quicker than the Gaussian distribution. Such a probability function will be of great use for describing the separation of categories in the pattern space when it is known with great certainty that the patterns cannot lie outside a certain set.

Beginning with equation 3.5

$$f_i(X) = \sum_{j=1}^{R} C(j/i)p(X/j)q_j$$

a discriminant function for this distribution can be derived in the same way as for the Gaussian case

$$p(X/j) = \{m_j\Gamma(n/2) \;/\; 2\Gamma(n/m_j)\pi^{\frac{1}{2}n}|\Sigma_j|\} \times$$

$$\times \exp\{-(X-M_j)^t\Sigma_j^{-1}(X-M_j)\}^{\frac{1}{2}m_j} \tag{3.14}$$

If we once again limit our discussion to the two category cases with

$C(i/i) = 0$ and use the form $f(X) = \log f_2(X) - \log f_1(X)$ with classification according to $f(X)<0$ for $X \varepsilon A_2$ and $f(X)>0$ for $X \varepsilon A_1$ we get

$$f(X) = \tfrac{1}{2}m_2(X-M_2)^t\Sigma_2^{-1}(X-M_2) - \tfrac{1}{2}m_1(X-M_1)^t\Sigma_1^{-1}(X-M_1) +$$

$$+ \log C(1/2) / C(2/1) + \log m_1/m_2 + \log q_1/q_2 +$$

$$+ \log \Gamma(n/m_2) / \Gamma(n/m_1) + \tfrac{1}{2}\log |\Sigma_2|/|\Sigma_1| \quad (3.15)$$

As before we can see the optimal solution for this distribution will be quadratic in X and if $\Sigma_1 = \Sigma_2$ the resulting function will be linear in X.

The training required for this distribution will differ from the Gaussian case only in that the values for m_1 and m_2 must be established.

3.3 Nonparametric Training Methods

In the study of nonparametric training, we consider classification problems in which an a priori knowledge of the distribution of pattern vectors in the pattern space is unnecessary. To train machines for classification under such conditions, we have only the knowledge which can be gained from the training samples to guide us. This type of training is probably more important than the parametric method because the parametric method begins by assuming some distribution and, consequently, the resulting classification cannot be any more accurate than the assumption.

We will assume in this section that a set of pattern vectors making up the training sample is available and the correct classification for each vector in this set is also available. These vectors

will be invariant and so we will be concerned with variations in the weight vector and the space from which this vector comes. We will not be interested in the pattern space. The Euclidean space, E^{n+1*}, which contains all the weight vectors, may be thought of as a dual space to the augmented pattern space. We will call this space the weight space. In the work that follows, the pattern space will also be assumed to be (n+1)-dimensional, i.e. the pattern vectors will all be assumed to be augmented as explained at the beginning of Chapter two.

As mentioned in equation 2.2, for classification to have any physical meaning, it is required that

$$A_1 \cap A_2 \cap \ldots \cap A_R = \emptyset$$

In this section, we will assume that this condition exists. This will enable us to assume the existence of a weight vector which will optimally dichotomize the pattern space provided it is used in conjunction with a proper Φ processor. Let the set of all these solution weight vectors be given by

$$V = \{W \epsilon E^{n+1*} : (W_i \cdot F(X)) > (W_i \cdot F(X)) \text{ for all}$$

$$X \epsilon A_i;\ i = 1, \ldots, R\}$$

The problem of training is then equivalent to choosing a W in V.

To begin the training process, an arbitrary (n+1)-dimensional vector is chosen for W. The vectors forming the training sample are then presented to the classifier one at a time and the response of the classifier is compared to the desired response. Each time the response is incorrect an adjustment is made on the weight vector. This procedure is continued with the training set being presented as many

times as necessary until the classifier can correctly classify all patterns in the training set. To explain the procedure more fully, let us consider a two category case. Suppose the pattern space contains two categories A_1 and A_2 with classification determined by $(W \cdot F(X)) > 0$ for $X \in A_1$ and $(W \cdot F(X)) < 0$ for $X \in A_2$. Consider the first time an error in classification occurs. We then have the condition $(W \cdot F(X)) < 0$ for $X \in A_1$ or $(W \cdot F(X)) > 0$ for $X \in A_2$. A correction must be made in W to produce a correct classification; in the first case W must be increased, and, in the second case W must be decreased. It is well known that the direction of maximum increase of a function is along the gradient while the direction of maximum decrease is along the negative of the gradient. Solving for the gradient of the discriminant function in the weight space we get

$$\nabla(W \cdot F(X)) = (\partial/\partial w_1, \partial/\partial w_2, \ldots, \partial/\partial w_{n+1})(W \cdot F(X))$$
$$= F(X) \qquad (3.16)$$

This means that no further mechanical calculations are required to find the gradient because F(X) is already available. This makes this method convenient for implementation.

We now know what the direction of the correction vector should be, we only need to know how large the correction should be. Let the fraction of F(X) which we use be c. For an incorrect guess on the k'th step of training the correction will then be

$$W(k+1) = W(k) \pm cF_k(X) \qquad (3.17)$$

For a correct guess, the weight vector remains unchanged:

$$W(k+1) = W(k) \qquad (3.18)$$

Many different suggestions for the value of c have been given. The

simpliest is to make c some arbitrary constant greater than zero. This leads to the fixed increment error correction procedure. Another method gives c the value

$$c = \lambda(W(k) \cdot F_k(X)) \; / \; |F_k(X)|^2 \qquad (3.19)$$

For $\lambda = 1$, this is the smallest number which must be added to W(k) to change the sign of $(W(k) \cdot F_k(X))$. For this value of λ this method is called the absolute correction rule.

To prove that such a training process does converge to a solution vector in V, the following modifications are introduced. Each time a pattern is presented and is correctly classified the weight vector remains unchanged. For this reason, we will consider a reduced training set which contains only those patterns for which the classifier gives an incorrect response. This means that a correction is made after each training pattern in this reduced set is presented to the classifier. Such a presentation and correction is called a step of learning. One further change is to negate all vectors F(X) derived from the vectors X of the training sample which belong to category A_2. This means $(W \cdot F(X)) < 0$ will be incorrect for all X and the correction required will be

$$W(k+1) = W(k) + cF(X)$$

These modifications do not change the basic training procedure, but will simplify our proof of convergence.

We can now say

$$W(k+1) = W(k) + cF_k(X)$$

so that

$$W(k+1) - W = W(k) - W + cF_k(X)$$

where W is a vector from the solution region V. Multiplying through by $-F_k(X)$ gives

$$W \cdot F_k(X) - W(k+1) \cdot F_k(X) = W \cdot F_k(X) - W(k) \cdot F_k(X) - c|F_k(X)|^2$$

Since c is positive

$$c|F_k(X)|^2 \geq 0$$

so

$$(W - W(k+1)) \cdot F_k(X) \leq (W - W(k)) \cdot F_k(X)$$

Now let

$$F_k(X) = W - W(k+1)$$

If no such vector exists in the training sequence we can introduce one into the sequence. We then have

$$(W - W(k+1)) \cdot (W - W(k+1)) \leq (W - W(k)) \cdot (W - W(k+1))$$

But

$$W(k+1) = W(k) + c(W - W(k+1))$$

$$= \{W(k) + cW\} \; / \; 1+c$$

Therefore

$$(W - W(k)) \cdot (W - W(k+1)) = (W - W(k)) \times$$

$$\times (W - \{W(k) + cW\} \; / \; 1+c)$$

$$= (1 \; / \; 1+c)(W - W(k)) \cdot (W - W(k))$$

Now $c>0$ so

$$(1 \; / \; 1+c)(W - W(k)) \cdot (W - W(k)) < |W - W(k)|^2$$

therefore

$$|W - W(k+1)|^2 < |W - W(k)|^2$$

This shows that such an error correction method does converge to a solution vector.

This training procedure can be extended to the multicategory case as well. In such a case, there would exist a set of R weight vectors to be adjusted. Classification in this case is determined by $x \in A_i$ if $F(X)\cdot W_i > F(X)\cdot W_j$ $(j = 1, 2, \ldots, R;\ j \neq i)$. The correction procedure in the multicategory case for misclassification by placing X in category h when it actually belongs to category i at the k'th step of training is

$$W_i(k+1) = W_i(k) + cF_k(X)$$
$$W_h(k+1) = W_h(k) - cF_k(x)$$
$$W_j(k+1) = W_j(k) \quad (j = 1, 2, \ldots, R;\ j \neq h,\ j \neq i)$$

As before, all the weight vectors remain unchanged if a pattern is correctly classified. In this correction procedure, c can be defined in the same way as in the single-category case. Nilsson[7] has shown that such an error-correction procedure does converge to a solution weight vector.

This particular approach to error correction is not a unique approach. Many authors have given different versions of this same basic approach, the original author being Rosenblatt[13]. Other nonparametric training methods have been proposed but none is as simple to implement. Two of these methods will be discussed and compared in the following chapter. An excellent review of the major training methods, both parametric and nonparametric, is given by Nagy[14] in a very recent paper. In this paper, he also gives a comparison of these methods. The error-correction method just described compares quite favorably, with the others.

CHAPTER 4 COMPUTER SIMULATION OF SOME TRAINING METHODS

4.1 Introduction

The current research in the area of training algorithms for pattern classifiers deals almost exclusively with nonparametric training. For realistic pattern recognition problems, nonparametric training is advantageous because it does not require a priori knowledge of the pattern sets. In practice, this knowledge will not likely be available. This chapter includes a review of two of the more recent nonparametric training methods along with the results of a computer simulation of a pattern recognition problem using them.

As was mentioned in section 3.1, nonparametric training procedures attempt to determine a weight vector, known as the optimal weight vector, which minimizes the risk or cost of misclassification. Without a complete knowledge of the pattern classes, this optimal vector cannot be calculated directly. For this reason, a search technique is introduced to find the minimum in the weight space of the cost function.

There exist many possibilities for deriving a training algorithm, primarily because the functional description of the cost of misclassification can be established in several different ways. Once this cost function is established, any one of several mathematical search techniques can be used to search the weight space for a weight vector to minimize this function. One of the most popular search techniques is the steepest descent method which minimizes a function along the negative of its gradient.

Many new and original methods appear in the literature almost continually. One completely new method has been proposed by

Chien[15]. This method assumes neither any knowledge of the distribution of pattern samples nor any knowledge of the number of categories present. While a critique of all the methods that have been proposed so far would seem to be desirable, such a study could not be attempted due to limitations of time and space. Instead, a comparative study of two of the algorithms presented in recent papers will be discussed in this chapter.

4.2 The Classification Problem

The idea of a computer simulation of training methods was provided by a paper by J.D. Patterson and B.F. Womack[16]. This paper contains a qualitative description of a nonparametric training algorithm and the results of a simulation study using this training method on four separate classification problems. All the classification problems treated in this paper consist of two sets of bivariate normally distributed pattern vectors with different means and variances. In checking the results of this paper, an ambiguity was found which will be explained later. For this reason it was decided to study the training problem presented in the paper to try to establish the possible source of this ambiguity.

Suppose that we have two categories from which the pattern vectors are derived. We will call these categories C_1 and C_2. The discriminant function will divide the pattern space into two half spaces:

$$V_1 = \{X: f(X) > 0\} \qquad V_2 = \{X: f(X) < 0\}$$

The decision rule is: if $X \varepsilon V_1$ decide $X \varepsilon C_1$, if $X \varepsilon V_2$ decide $X \varepsilon C_2$. The cost of misclassifying a pattern from C_α by placing it in C_β will be given by $C(\beta/\alpha)$, $(\alpha,\beta = 1, 2)$.

The optimum discriminant function for dichotomizing a pattern

space containing two categories of Gaussian distributed pattern vectors can be calculated directly by using equation 3.8. The resulting discriminant function is called the Baye's discriminant function for a Gaussian distribution. The means and variances for the four problems are given in Table 4.1.

In the following work, the two algorithms used for training will be compared. One of the most obvious ways of comparing the two methods is to compare the number of errors each one produces on a given sample set after training. The way to produce the minimum number of errors is to set the costs of misclassification equal and the probability of occurrence of the sample sets equal. Substituting these values into equation 3.8 we get the following discriminant functions:

$$f(X) = (-4x_1 - x_2 + 5) / 2 \qquad (4.1)$$

$$f(X) = 2.69 - (3x_2^2) / 32 - 2x_1 \qquad (4.2)$$

$$f(X) = (22.18 - 15x_1^2 + 12x_2^2) / 32 \qquad (4.3)$$

$$f(X) = (44.35 - 12x_1^2 - 3x_2^2) / 32 \qquad (4.4)$$

The Baye's solution with $C(1/2) = C(2/1)$ and $q_1 = q_2$ produces an equal probability of misclassification from either category for a continuous distribution. This probability can be obtained by integrating one distribution over the space on which it can produce a misclassification or taking the integral over the space on which it is correctly classified and subtracting this from 1. Choosing for the sake of simplicity the distribution of category one we get

$$P(\text{Error}) = \int_{V_2} p_1(X)\, dX$$

$$= 1 - \int_{V_1} p_1(X)\, dX$$

PROBLEM	MEANS	VARIANCES	P(ERROR) FROM EXACT BAYE'S RULE SOLUTION	P(ERROR) FROM ESTIMATED BAYE'S RULE SOLUTION	P(ERROR) FROM RESULTS OF ADAPTIVE PROCEDURES
1	$M_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $M_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$	$\Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$ $\Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$	.1317	.095	.085
2	$M_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $M_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$	$\Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$ $\Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 16 \end{bmatrix}$	.1307	.105	.110
3	$M_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $M_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$ $\Sigma_2 = \begin{bmatrix} 16 & 0 \\ 0 & 1 \end{bmatrix}$	.0989	.150	.230
4	$M_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $M_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$ $\Sigma_2 = \begin{bmatrix} 4 & 0 \\ 0 & 16 \end{bmatrix}$	.1574	.220	.270

TABLE 4.1 SUMMARY OF PATTERSON AND WOMACK'S RESULTS TOGETHER WITH THE PROBABILITY OF ERROR {P(ERROR)} FROM THE EXACT BAYE'S RULE SOLUTION

After calculating this integral for the four cases*, we get the solutions given under the heading 'P(ERROR) FROM EXACT BAYE'S RULE SOLUTION' in Table 4.1.

At this point, we have a disagreement with the authors of the aforementioned paper. In their results, they have listed the probabilities of misclassification due to the results of their adaptive training and also due to an estimated Baye's Rule classification. These are given in Table 4.1. As can be seen these values differ from those obtained by using exact Baye's rule solution. The results for cases one and two are especially interesting since their estimation procedures have yielded probabilities lower than the theoretically optimum. For this reason, we decided to apply the training algorithm given in their paper to the first categorization problem to determine the accuracy of the remainder of the paper.

The training problem is to find a decision boundary between two normally-distributed pattern sets which minimizes the classification error. Classification takes place according to $X \varepsilon C_1$ if $f(X) > 0$ and $X \varepsilon C_2$ if $f(X) < 0$. The theoretically optimal decision boundary is given by setting $f(X)$ from equation 4.1 equal to zero. For the simulation study, the continuous Gaussian distributions are approximated by 50 pattern vectors from each distribution. The method used for obtaining a Gaussian distribution of these pattern vectors is one recommended by IBM for such purposes[19]. Assume the points z_i are uniformly distributed in the interval (0,1). An approximate normal function with 0 mean and a standard deviation of 1 obtained from the z_i is

---

*See Appendix A for detailed calculations.

$$Y' = \left(\sum_{i=1}^{N} z_i - N/2 \right) / \sqrt{N/12} \qquad (4.6)$$

This can be changed to a normal function of mean m and standard deviation σ by taking

$$Y = \sigma Y' + m \qquad (4.7)$$

The z_i were obtained by using the random number generator. After a number of trials N was chosen to be 179. To approximate a uniform distribution in (0,1), 179 numbers between 0 and 10,000 were randomly chosen and each was divided by 10,000. This gave the z_i from which the Y was calculated.

A univariate normal distribution has a probability of .683 of the variable lying within the standard deviation of the mean. For a bivariate normal distribution, this probability is $(.683)^2$ or .4665. In the sample patterns used, 44% of category 2 and 48% of category one lie within the standard deviation of the mean. The sample patterns seem to be fairly accurate representations of the normal distributions for which they were intended.

One further check was made by applying the optimal discriminant function given by equation 4.2 to this simulated pattern space. This resulted in fifteen errors, eight from category one and seven from category two. This means an overall probability of error of .15 compared with .1317 for the theoretical probability.

4.3 Mean-Square-Error Classifier

In this section, we will discuss the training algorithm proposed by Patterson and Womack[16]. In their paper, they have presented the criterion for classification and given the results of a simulation

study. The search technique used, which they have called 'pattern search', is mentioned but not discussed in their paper. For this reason, the steepest descent search technique is applied to the criterion given in the paper. This algorithm will then be applied to the first classification problem given in section 4.2.

The criterion for classification can be established by considering the discriminant function as a mapping from the pattern space to the real line which may be considered the 'decision space.' In performing this mapping it is desirable to map the points from C_1 as near as possible to some point K_1 which is a finite distance from the point K_2 to which the points from C_2 map. Since we desire $f(X) > 0$ for $X \varepsilon C_1$, the constant K_1 will be positive and similarily we see K_2 will be negative. To weight the classification according to the predetermined costs of misclassification, let $K_1 = C(2/1)$ and $K_2 = -C(1/2)$. One method of measuring the effectiveness of a discriminant function would then be to measure the error produced in this mapping. For this, we use the mean-square-error M:

$$M = \overline{\{f(X) - C(2/1)\}^2}^{(1)} + \overline{\{f(X) + C(1/2)\}^2}^{(2)} \qquad (4.8)$$

where $\overline{}^{(i)}$ denotes averaging over category i. Taking an estimate of the probability of occurrence of the two populations to be q_1 and q_2 and the number of samples from each population to be N_1 and N_2, equation 4.8 may be written as

$$M = (q_1/N_1) \sum_{\alpha=1}^{N_1} \{f(X_\alpha) - C(2/1)\}^2 +$$

$$+ (q_2/N_2) \sum_{\beta=1}^{N_2} \{f(X_\beta) + C(1/2)\}^2 \qquad (4.9)$$

The steepest descent method will be applied to the mean-square-error criterion to develop a search algorithm. The mean-square-error is a functional of the discriminant function which in turn is a function of the pattern vector and the weight vector. The mean-square-error is changed by successive adjustments of the weight vector. To minimize the mean-square-error in a minimum of time, we need to know its gradient, ∇M, since the negative of the gradient will give the direction of maximum decrease or steepest descent. If a series of infinitesimal steps were taken in the direction of the negative gradient the change in M would always be in the direction of maximum decrease. Since this would require an infinite number of steps to move a finite distance, some finite distance λ must be chosen in the direction of $-\nabla M$. Denoting $-\nabla M$ by s we get

$$M(W+\lambda s) = (q_1/N_1) \sum_{\alpha=1}^{N_1} \{f(W+\lambda s \cdot X_\alpha) - C(2/1)\}^2 + \\ + (q_2/N_2) \sum_{\beta=1}^{N_2} \{f(W+\lambda s \cdot X_\beta) + C(1/2)\}^2 \tag{4.10}$$

Solving for the gradient* we get

$$s = (q_1/N_1)\{(\sum_{\alpha=1}^{N_1} A(X_\alpha)) \times W - \sum_{\alpha=1}^{N_1} F(X_\alpha) \times C(2/1)\} + \\ + (q_2/N_2)\{(\sum_{\beta=1}^{N_2} A(X_\beta)) \times W + \sum_{\beta=1}^{N_2} F(X_\beta) \times C(1/2)\} \tag{4.11}$$

where F(X) is the augmented Φ function of the pattern vector and A(X) is the matrix made by taking the outer product of F(X) with itself.

---

*See Appendix B for calculations.

Now let us define

$$\lambda_1 = (q_1/N_1) \sum_{\alpha=1}^{N_1} \{ f(s \cdot X_\alpha) [f(W \cdot X_\alpha) - C(2/1)] \}$$

$$\lambda_2 = (q_2/N_2) \sum_{\beta=1}^{N_1} \{ f(s \cdot X_\beta) [f(W \cdot X_\beta) + C(1/2)] \}$$

In terms of these new variables, the minimum of $M(W + \lambda s)$ is given by*

$$\lambda = -(\lambda_1 + \lambda_2) \Big/ (q_1/N_1) \sum_{\alpha=1}^{N_1} f(s \cdot X_\alpha)^2 + (q_2/N_2) \sum_{\beta=1}^{N_2} f(s \cdot X_\beta)^2 \qquad (4.12)$$

These results can now be used for nonparametric training of a pattern classifier. The various steps are:

1) The $N_1 + N_2$ samples together with their correct classifications are given to the classifier which makes the indicated calculations on these vectors and produces a value for λ and s.
2) These values are then used to change the weight vector W to a new value of $W + \lambda s$.
3) The process is then repeated using the $N_1 + N_2$ samples each time until the mean-square-error is near enough to zero.

A schematic of this scheme is given in Figure 4.1. For a definition of the terms used see the calculations in the appendix.

Because the distributions of pattern vectors to be classified are Gaussian, a quadratic Φ processor is used in the simulation study. This also allows a comparison of the simulation results with

---

*See Appendix B for calculations.

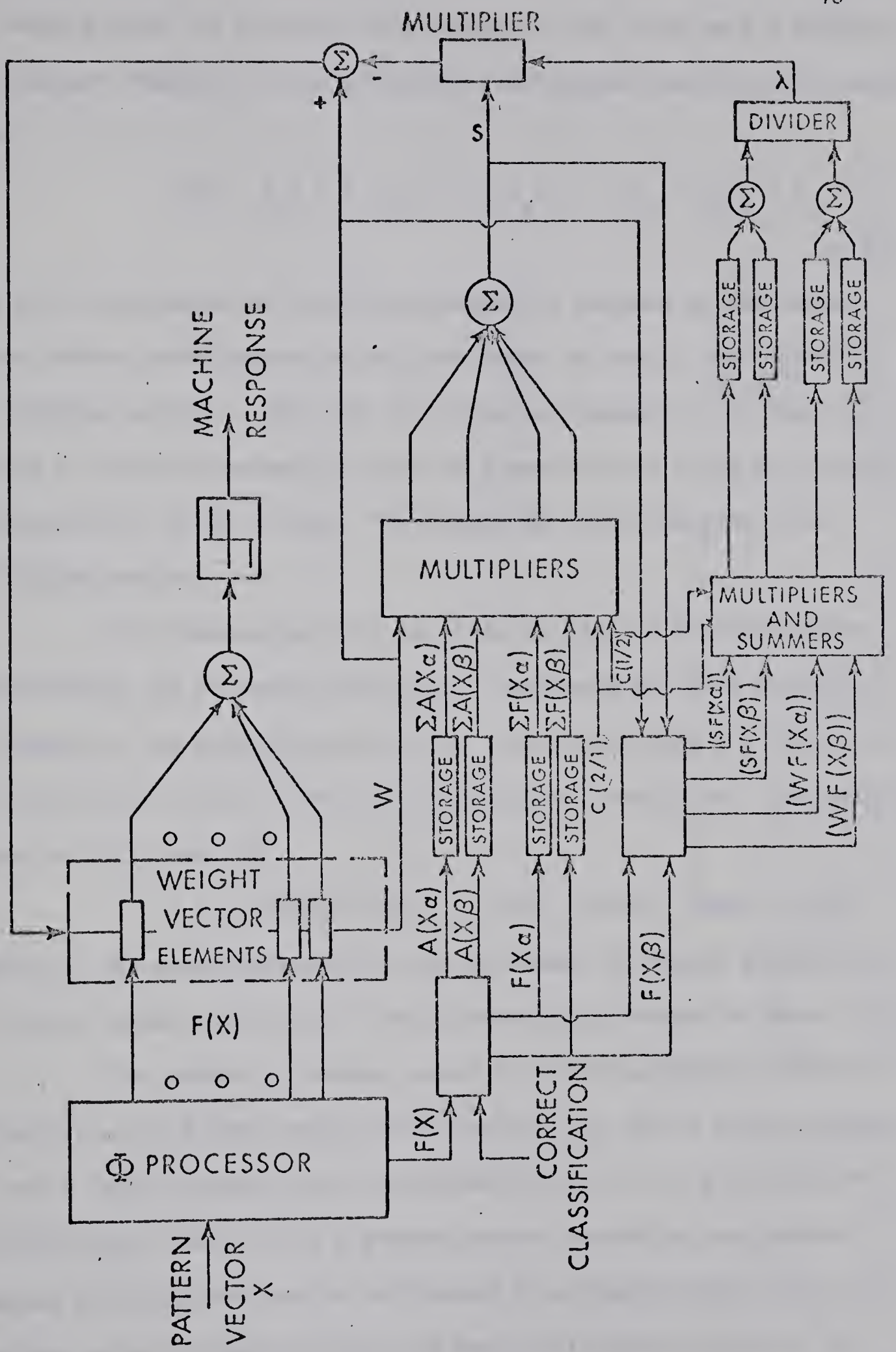


Fig. 4.1 MEAN-SQUARE-ERROR PATTERN CLASSIFIER

the results given by Patterson and Womack as they also used a quadratic discriminant function. This gives the discriminant function the general form:

$$f(X) = w_1 x_1^2 + w_2 x_2^2 + w_3 x_1 x_2 + w_4 x_1 + w_5 x_2 + w_6 \tag{4.13}$$

No a priori knowledge of the distributions is assumed by the weight vector and so its elements are all set equal to one at the outset of the training process. The cost functions are assumed to be equal in section 4.2 and are assumed to take on some constant value so they are arbitrarily set equal to one. The system is then simulated on an IBM APL\360 system.

The training process as given on page 45 is repeated one hundred times and the mean-square-error decreased for each iteration. The change in the mean-square-error is found to be from M = 597.6 for W = 1,1,1,1,1,1 to M = 1.089 after one hundred iterations. The weight vector at this point is

W = -.2253, -.03314, -.06276, .2987, .09026, .8023

A graph of the mean-square-error and the number of errors produced by the weight vector at the end of each iteration is shown in Figure 4.2.

The number of errors caused by the discriminant function derived from the weight vector after training is 36, 28 from category two and 8 from category one. The probability of error arising from the use of this function on a pattern space containing continuous Gaussian distributions can be calculated in a similar way to the calculations given in section 4.2. One major difference, however, is that the probabilities of misclassification from the two categories

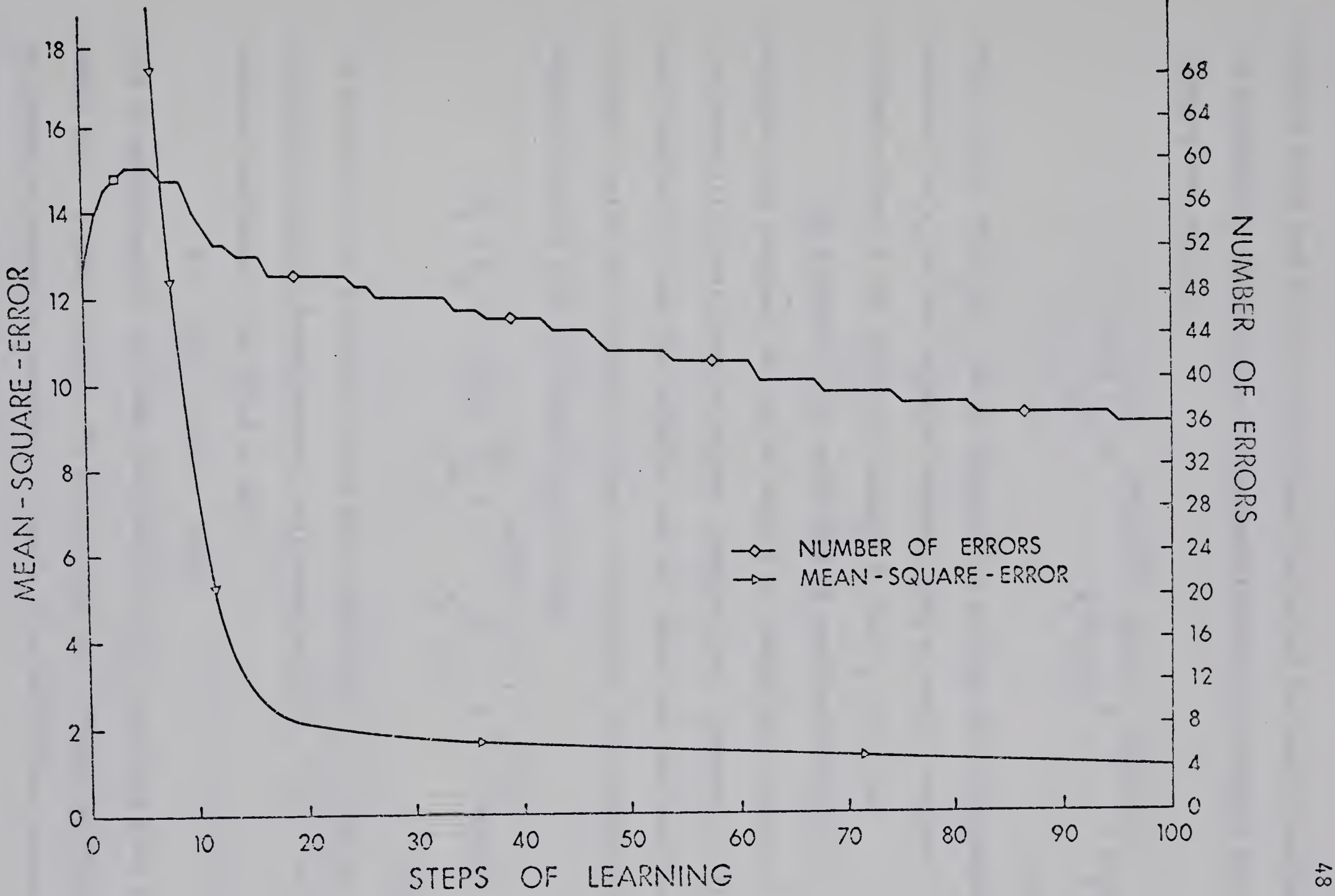


FIGURE 4.2 MEAN-SQUARE-ERROR AND NUMBER OF ERRORS FOR MEAN-SQUARE-ERROR CLASSIFIER.

are not equal and so the integral must be solved for each distribution. The probability of error due to misclassification of patterns from category two will be

$$P_2 = q_2\{(1/4\pi) \int_{x_1=a_1}^{a_2} e^{-(x_1-2)^2/2} \int_{x_2=f_1(x_1)}^{f_2(x_1)} e^{-(x_2-2)^2/8} dx_2 dx_1\} \tag{4.14}$$

where $f_1(x_2)$ and $f_2(x_2)$ can be obtained by solving the discriminant function for x_1 at the decision boundary, and a_1 and a_2 are the extreme values of the decision boundary in the x_1 direction.

The probability of error due to misclassification of patterns from category one can be solved by taking the integral of its probability distribution outside category one. However, it is much simplier to take the integral over all space and subtract the integral inside this category. Since the integral of a probability density function over all space is one we get

$$P_1 = q_1\{1 - (1/4\pi) \int_{x_1=a_1}^{a_2} e^{-x_1^2/2} \int_{x_2=f_1(x_1)}^{f_2(x_1)} e^{-x_2^2/8} dx_2 dx_1\} \tag{4.15}$$

The probability of error when using this discriminant function on continuous Gaussian distributions will then be $P = P_1 + P_2$. For the example considered, P is found to be*

$$P = .0743 + .2925 = .3668$$

This is approximately the same conclusion as that resulting from the sample set. A graph showing the points from each category as well as the decision boundaries produced by Baye's Rule and by the training process is given in Figure 4.3.

*See Appendix C for calculations.

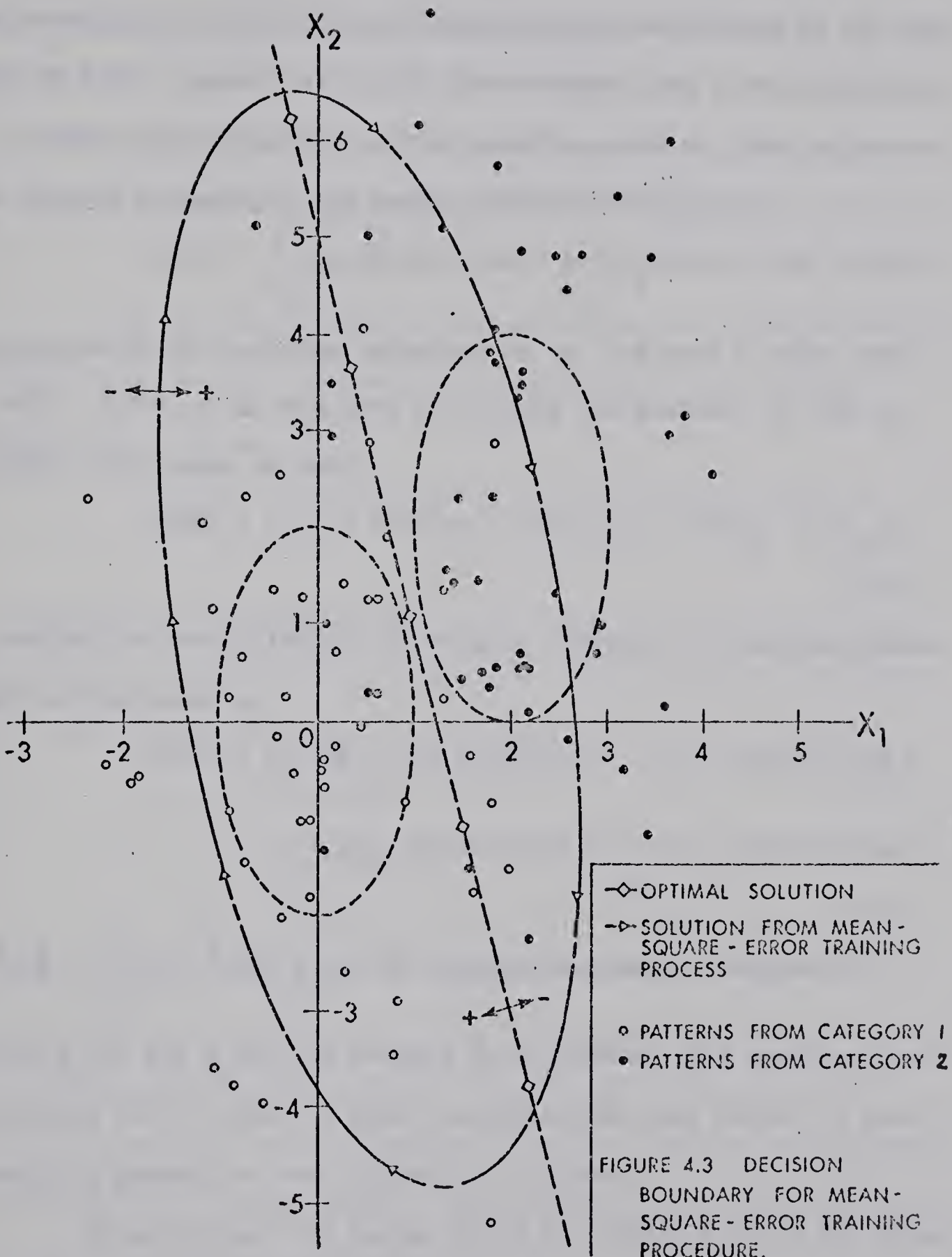


FIGURE 4.3 DECISION BOUNDARY FOR MEAN-SQUARE-ERROR TRAINING PROCEDURE.

4.4 Probabilistic - Descent Classifier

In this section, a nonparametric training method proposed by Amari[17] is discussed. The criterion used in this training procedure is the probability of an error in classification multiplied by the cost of such an error. Amari calls this "the average risk of misclassification". Using the terminology of the preceding section, this criterion can be defined in terms of the weight vector W as follows:

$$R(W) = \int_{V_1} q_2 p_2(X) C(2/1) dX + \int_{V_2} q_1 p_1(X) C(1/2) dX \quad (4.17)$$

The objective of the training procedure is to find that W which minimizes R(W). This can be achieved by setting the gradient of R(W) in the weight space equal to zero.

$$\nabla R(W) = 0 \qquad \nabla = (\partial/\partial w_1, \partial/\partial w_2, \ldots, \partial/\partial w_n, \partial/\partial w_{n+1}) \quad (4.18)$$

This gradient is very difficult to obtain. However, it has been given by Amari in his paper as

$$\nabla R(W) = (1/\|W\|) \int_D X\{q_2 p_2(X) C(2/1) - q_1 p_1(X) C(1/2)\} dX + \int_{V_1} \nabla\{q_2 p_2(X) C(2/1)\} dX + \int_{V_2} \nabla\{q_1 p_1(X) C(1/2)\} dX = 0 \quad (4.19)$$

where $\|W\| = (\sum_{i=1}^{n} w_i^2)^{\frac{1}{2}}$ and D is the decision boundary. Because the values of $P_1(X)$ and $P_2(X)$ are assumed to be unknown this equation cannot be solved for W. Even if these probabilities were known the solution would in general be very difficult to obtain.

If we now use the values given in section 4.3 for the parameters in equation 4.19, the last two integrals are zero, since q_1, q_2, C(1/2) and C(2/1) are constants and $p_1(X)$ and $p_2(X)$ are functions only

of the pattern vector X. The first integral is, however, much more difficult to work with than the following two. For this reason, Amari assumes the cost function to be a function of the distance from the decision boundary in such a way that its value on the boundary is zero. For this special case, the first integral in equation 4.19 is zero and the optimal weight vector is given by the solution to the equation

$$\nabla R(W) = \int_{V_1} \nabla\{q_2 p_2(X) C(2/1)\} dX + \int_{V_2} \nabla\{q_1 p_1(X) C(1/2)\} dX = 0 \tag{4.20}$$

The only part of the integrands of these integrals which is a function of W will then be the cost function C(1/2) and C(2/1). Therefore

$$\nabla R(W) = q_2 \int_{V_1} p_2(X) \nabla C(2/1) dX + q_1 \int_{V_2} p_1(X) \nabla C(1/2) dX = 0 \tag{4.21}$$

For a nonparametric training method, the weight vector, W, is changed by an amount δW every time a pattern vector is classified. In Amari's training method this change in W is given by

$$\delta W_i = E\, H(X_i, W_i) \tag{4.22}$$

where

$$H(X, W) = \begin{cases} H_1(X, W) & \text{when } f(X) < 0 \text{ and } X \in A_1 \\ H_2(X, W) & \text{when } f(X) > 0 \text{ and } X \in A_2 \\ 0 & \text{when X is correctly classified} \end{cases}$$

and E is a positive-definite matrix. This training procedure is guaranteed to converge if $H_1 = -\nabla C(1/2)$ and $H_2 = -\nabla C(2/1)$. The proof of this statement is quite simple. We know $\delta W = EH$ for each misclassified pattern, so the average change in W is

$$\delta W = Eq_1 \int_{V_2} p_1(X)\ H_1 dX + Eq_2 \int_{V_1} p_2(X)\ H_2 dX$$

$$= -Eq_1 \int_{V_2} p_1(X) \nabla C(1/2) dX - Eq_2 \int_{V_1} p_2(X) \nabla C(2/1) dX$$

$$= -E\ \nabla R(W)$$

We also know

$$\partial R(W) \ / \ \partial W = \nabla R$$

If the values of the elements of E are sufficiently small we can approximate ∂W by δW. We can then say, the increment in the risk function is

$$\delta R(W) = \delta W^t\ \nabla R$$

for a change of δW in W. Using the value just solved for δW the average increment in the risk function is

$$\delta R(W) = \delta W^t\ \nabla R = -\nabla R^t(W)\ E\ \nabla R(W)$$

Since E is positive definite

$$\delta R(W) \leq 0$$

But R(W) will be positive since the probability functions are always positive and the loss associated with misclassification will always be positive. Therefore, the average change in R(W) will cause a decrease in the risk until the optimum W is reached for which

$$\delta R(W) = 0$$

Since this decrease in the risk function occurs only as an average and not on each individual step, Amari has called this method the probabilistic-descent method.

Amari has further improved this training theorem by incorporating the ability of the system to learn the learning rule. This is achieved by making the values of E adaptable. The values of the elements of E determine what size the increment δW will be. When the

weight vector is far from being optimal it is desirable to make W large. However, as W approaches the optimal value, it becomes desirable to make the change small. For this reason, we introduce a change in E of ΔE for every time a change in W occurs. The value of ΔE takes on is given by

$$\Delta E = \gamma\ H(X,\ W)\ H^{t}(X',\ W')$$

when the pattern X' is misclassified by using the weight vector W', where X is the previously misclassified pattern and γ is a positive constant. Amari shows this change in E will produce relatively large changes in W in the direction of the gradient of the risk function when W is far from optimal, but tends to reduce the absolute value of δW as W approaches the optimal value. This technique would, therefore, increase the accuracy of the training procedure.

For the cost function Amari suggests using the form

$$C = |f(X)| \ / \ \|W\| \tag{4.23}$$

This function will assign a small cost to misclassified patterns near the decision boundary and much greater cost to those far from the boundary. However, we assumed at the outset of the present discussion that a constant loss would be attributed to any misclassification. To approximate such a situation with a distance-cost function, the following cost function can be used:

$$C(2/1) = C(1/2) = \{|f(X)| \ / \ \|W\|\}^{p} \tag{4.24}$$

where $p << 1$. Using this function, the cost of misclassification will be close to one for any misclassified pattern.

Having found an appropriate cost function, we can now find H(X, W). Since we are considering misclassified patterns we have

$$C(1/2) = \{-f(X)\}^p \; / \; \|W\|^p$$

so

$$\nabla C(1/2) = p\{-f(X) \; / \; \|W\|\}^{p-1} \; \{(-X \; / \; \|W\|) - f(X) \; \nabla(1 \; / \; \|W\|)\} \tag{4.25}$$

$$\|W\| = (w_1^{\,2} + w_2^{\,2} + \ldots + w_n^{\,2})^{\frac{1}{2}}$$

Therefore

$$\nabla(1 \; / \; \|W\|) = -\tfrac{1}{2}(w_1^{\,1} + w_2^{\,2} + \ldots + w_n^{\,2})^{-3/2} \times$$

$$\times \; (2w_1, \; 2w_2, \; \ldots, \; 2w_n)$$

$$= -W \; / \; \|W\|^3$$

So

$$\nabla C(1/2) = p\{-f(X) \; / \; \|W\|\}^{p-1} \; \{(-X \; / \; \|W\|) + (W \; / \; \|W\|^3)\} \tag{4.26}$$

Similarily

$$C(2/1) = \{+f(X)\}^p \; / \; \|W\|^p$$

so

$$\nabla C(2/1) = p\{+f(X) \; / \; \|W\|\}^{p-1} \; \{(X \; / \; \|W\|) - Wf(X) \; / \; \|W\|^3\} \tag{4.27}$$

Setting $H_1 = -\nabla C(1/2)$ and $H_2 = -\nabla C(2/1)$, we can now establish the nonparametric training scheme.

This training method is then applied to the same problem mentioned previously and is simulated on the IBM APL\360 system using the same set of training samples. The matrix E must be positive definite and so for simplicity it was initially set equal to the identity matrix. The values of H(X,W) and H(X', W'), however, were found to be less than one and so γ was arbitrarily set at .5. To choose an appropriate value for p some calculations were performed. After some trial iterations,

it is found that the value of $|f(X)|$ will almost certainly lie in the range $10^{-4} < |f(X)| < 100$ and the value of $\|W\|$ will almost certainly lie in the range $1 < \|W\| < 10$. This means $|f(X)| / \|W\|$ will with a very high degree of certainty, lie in the range $10^{-5} < |f(X)| / \|W\| < 100$. p should now be chosen so that C(1/2) and C(2/1) will be close to one for all values of W and X. A choice of p = .05 is made which makes the range of the cost function .562 < C(1/2) = C(2/1) < 1.259 with C(1/2) and C(2/1) being within 25% of one for almost all values of W and X.

Using these values for the constants for the problem and setting the initial weight vector at W = (1,1,1,1,1,1,), the simulation study is attempted. The results are given by curve (1) of Figure 4.4.*

As can be seen from the figure, the convergence rate is very fast and the weight vector approaches the optimal but the solution changes dramatically during the 16th step of learning. This sudden change has been traced in the program and is found to occur on a misclassification of a pattern from category two. The discriminant function for this pattern vector takes on the value .0142. The weight vector at this point is W = (2.208, -.552, -1.194, 3.764, -1.021, 2.696). This gives the norm a value of $\|W\|$ = 4.673 yielding a cost of C(2/1) = .7484. However,

---

*This graph differs slightly from Figure 4.2. In the mean-square-error classifier, the weight vector was changed only after all 100 patterns had been received and the appropriate calculations made. In the probabilistic-descent classifier the weight vector was changed each time an error in classification resulted. However, it would have required far more computer time to stop after each correction and check to see how many of the 100 patterns would be misclassified by this new weight vector. For this reason, the weight vector resulting after the 100 patterns had all been analyzed was the only one checked for the errors it produced. This means the 'steps of learning' referred to in Figure 4.4 are not the number of times the weight vector has been changed but rather the number of times the 100 patterns have been analyzed.

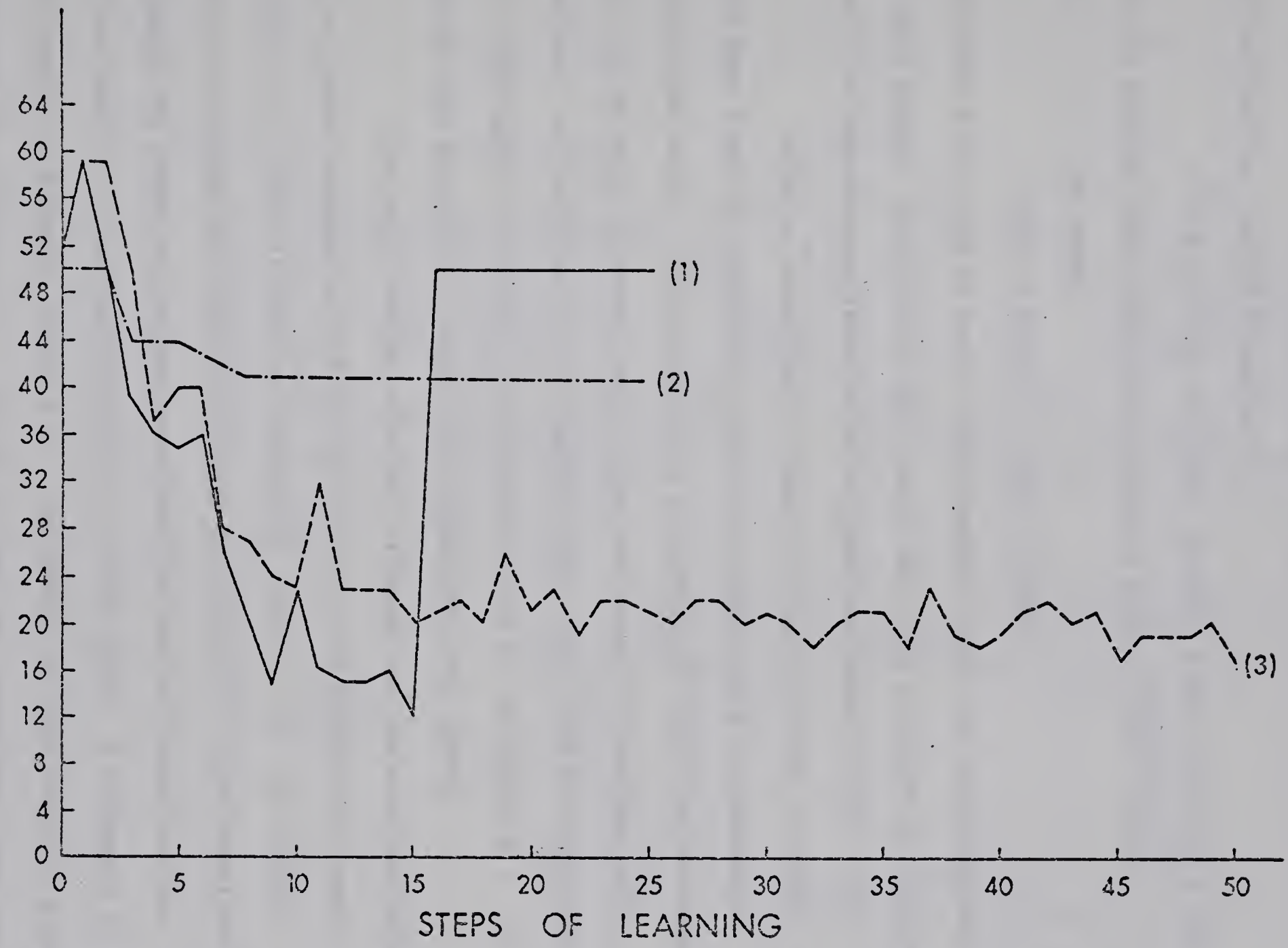


FIGURE 4.4 NUMBER OF ERRORS FOR PROBABILISTIC - DESCENT CLASSIFIER.

$$H_2 = -\nabla C(2/1) = p\{f(X) / \|W\|\}^{p-1} \{(X / \|W\|) - Wf(X) / \|W\|^3\}$$

$$= (-11.64, -35.25, -20.26, -5.525, -9.625, -2.627)$$

and multiplying this by E results in

$$\delta W = (-5.901, -45.218, -34.878, 5.647, -22.778, 5.285).$$

The big numbers result from multiplying by $\{f(X) / \|W\|\}^{p-1}$ where $p-1 = -.95$. This means

$$\{f(X) / \|W\|\}^{p-1} \simeq \{f(X) / \|W\|\}^{-1}$$

and since $f(X) / \|W\|$ is small (approximately .003) its reciprocal will be very large. This problem will result any time a pattern is misclassified by a discriminant function which gives the result $|f(X)| << 1$.

After a catastrophic change in W such as this, it would take many steps of learning to reduce the values of W to reasonable proportions again. To overcome this problem it is felt that increasing the size of each increment would allow the system to overcome such a problem. So the value of γ is changed from .5 to 5 and the simulation is begun again. The result is shown by curve (2) of Figure 4.4. It can be seen that such a change only degenerated the system.

The most obvious solution, although not the most desirable, is to ignore elements of the pattern sample which produce the result $|f(X)| << 1$. This is done by testing for $|f(X)| < .5$. For pattern vectors yielding this value, no change is made in W. This technique reduces the number of patterns available for use in classification. γ is set back to .5 and the simulation is tried again. A schematic of the system is given in Figure 4.5. The results of this test are given by curve (3) of Figure 4.4. As can be seen the convergence rate has been decreased by putting this test in the system.

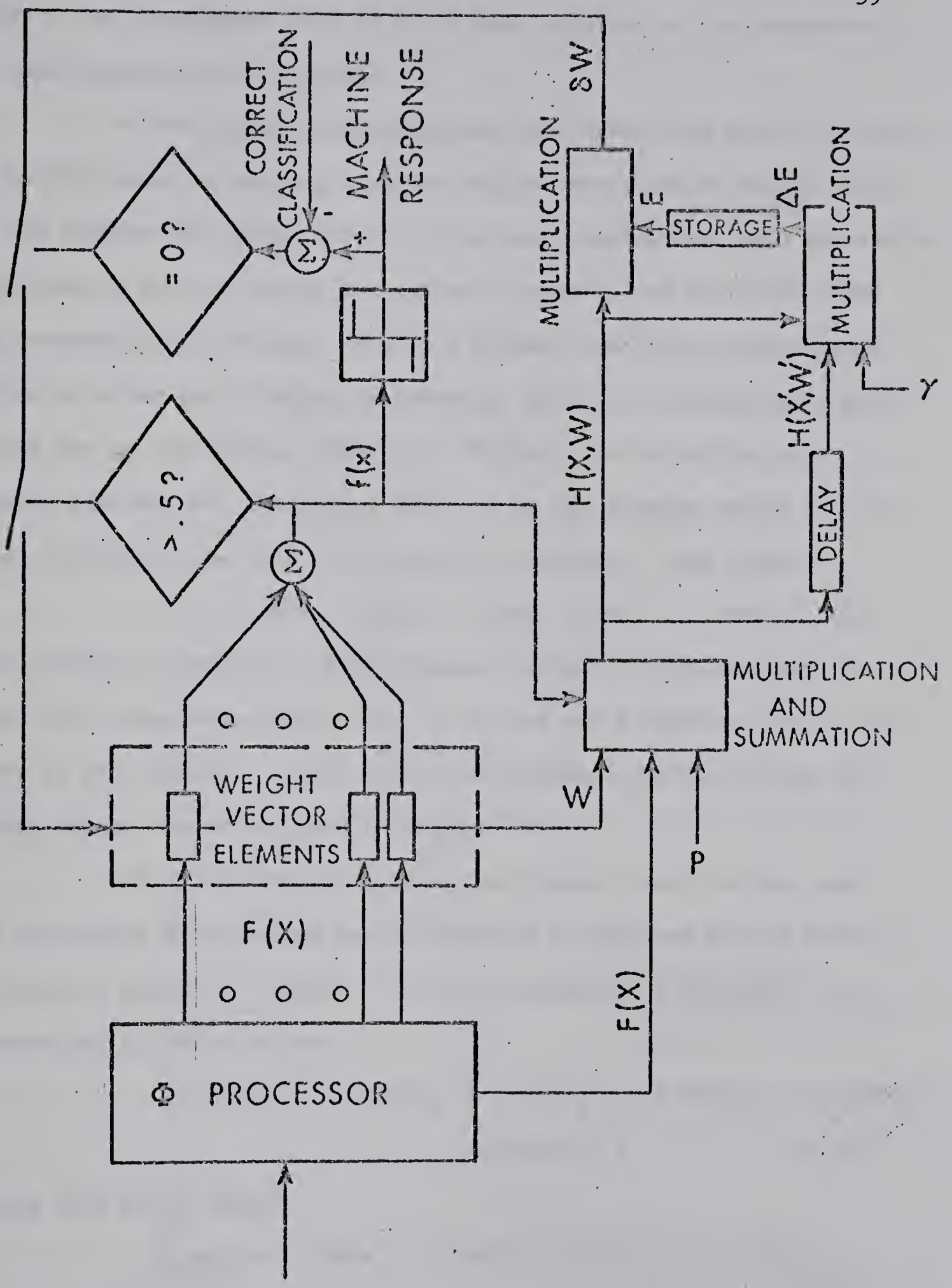


Fig. 4.5 PROBABILISTIC - DESCENT PATTERN CLASSIFIER

However, the convergence rate is still much superior to the results of the mean-square-error classifier.

As has already been mentioned, the error rate given in Figure 4.5 is the result of testing only the weight vector which results after all 100 samples have been tested. It is also evident that each successive weight vector may not reduce the number of errors, but the error rate does decrease as an average. For this reason, the final weight vector arrived at after the 50 steps of learning will not necessarily be the closest one to the optimal solution. The solution vector for this training process is, therefore, taken to be the average vector of all those realized in the last three steps of learning. This vector

$$W = -3.2896, -.04201, -.7481, 1.8927, -1.1685, 5.5026$$

which produces 16 errors, 9 from category one and 7 from category two. In the last three steps, there are 49 errors which averages out to 16.3 errors in 100 samples. A plot of the discriminant function using this average weight vector is given in Figure 4.6.

The effectiveness of this discriminant function when used on a continuous distribution can be measured in the same way as before. The solution graphed in Figure 4.6 is one section of a hyperbola. It is described by the equation

$$-3.2896x_1^2 - .04201x_2^2 - .7481x_1x_2 + 1.8927x_1 - 1.1685x_2 + 5.5026 = 0 \qquad (4.28)$$

Solving this for x_2 gives

$$f_1(x_1) = (.7481x_1 + 1.1685 + g(x_1)^{1/2}) \,/\, 2(-.04201) \qquad (4.29)$$

where

$$g(x_1) = (.7481x_1 + 1.1685)^2 + 4(.04201)(1.8927x_1 + 5/5026 - 3.2898x_1^2)$$

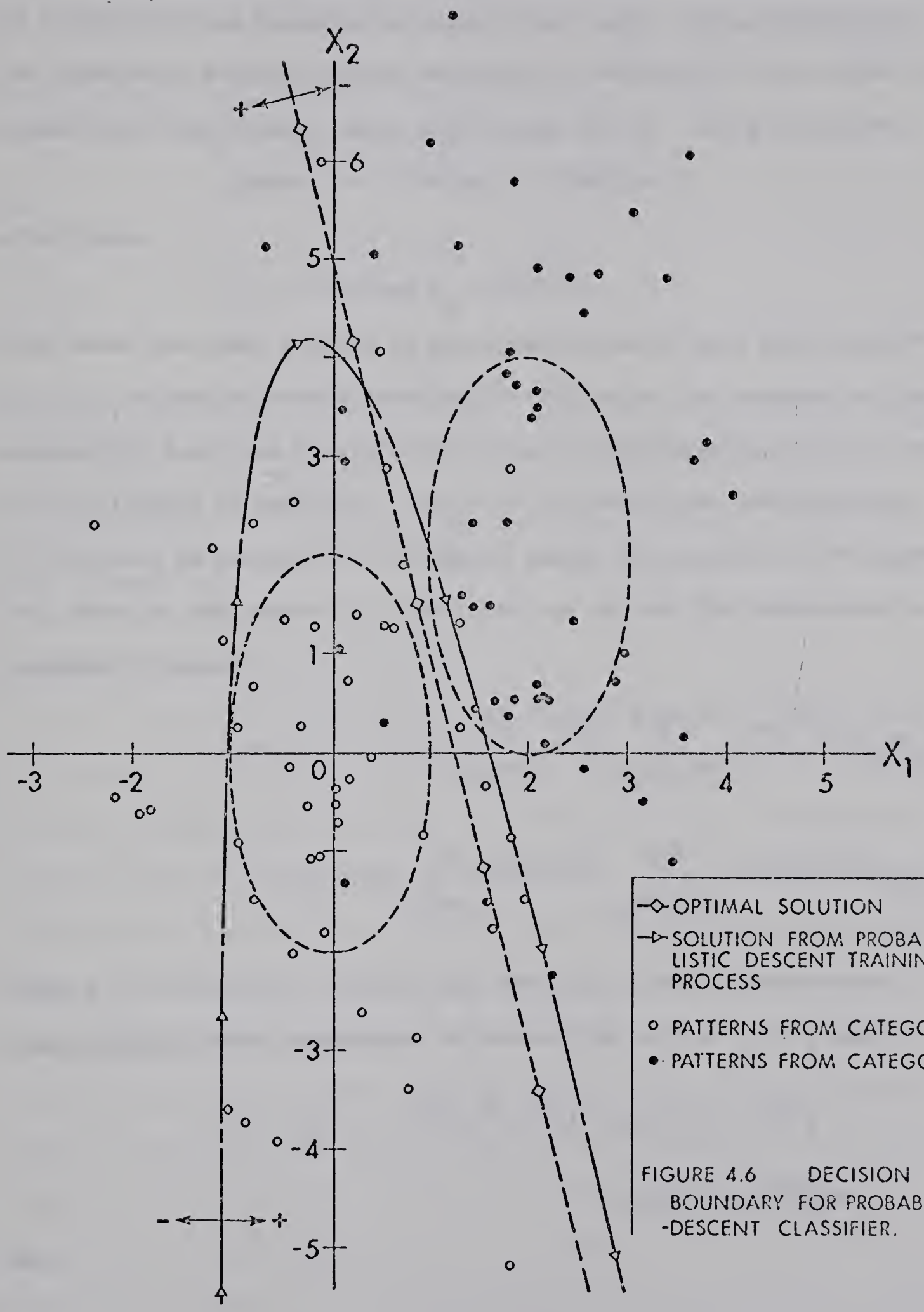


FIGURE 4.6 DECISION BOUNDARY FOR PROBABILISTIC -DESCENT CLASSIFIER.

It also gives $f_2(x_1)$ which is the same except that a negative instead of a positive sign precedes the square-root term. The extremities of the hyperbolic sections can be determined by setting the term under the square-root sign equal to zero and solving for x_1. Doing this gives

$$.00689x_1^2 + 2.06636x_1 + 2.29005 = 0$$

which gives

$$x_1 = -1.112 \text{ and } x_1 = -299.70.$$

This means the other section of the hyperbola never goes more positive in the x_1 direction than approximately -300 and so the integral of the probability functions inside this section of the hyperbola will be zero to five figures of accuracy. Therefore, to deduce the probabilities it will only be necessary to integrate inside the section of the hyperbola shown in the Figure 4.6. For this, we can use the form given in appendix C, namely,

$$P = q_1\{1 - (1 / 4\pi) \int_{x_1=a_1}^{a_2} e^{-\frac{1}{2}x_1^2} \int_{x_2=f_1(x_1)}^{f_2(x_1)} e^{-x_2^2/8}\, dx_2 dx_1\} +$$

$$+ q_2\{1/4\pi \int_{x_1=a_1}^{a_2} e^{-\frac{1}{2}(x_1-2)^2} \int_{x_2=f_1(x_1)}^{f_2(x_1)} e^{-(x_2-2)^2/8} dx_2 dx_1\}$$

where $a_1 = -1.112$, $a_2 = \infty$ and $f_1(x_1)$ and $f_2(x_1)$ are as given above. From appendix C this expression can be written as $P = P_1 + P_2$ where

$$P_1 = q_1\{1 - 1/\sqrt{8\pi} \int_{a_1}^{a_2} e^{-\frac{1}{2}x_1^2} \left[\text{erf}\left[f_2(x_1)/\sqrt{8}\right] - \text{erf}\left[f_1(x_1)/\sqrt{8}\right]\right] dx_1\}$$

and

$$P_2 = q_2\{1/\sqrt{8\pi} \int_{a_1}^{a_2} e^{-\frac{1}{2}(x_1-2)^2} \left[\text{erf } [f_2(x_1) - 2/\sqrt{8}\,] - \text{erf } [f_1(x_1) - 2/\sqrt{8}]\right] dx_1\}$$

These integrals were solved by computer methods to give

$$P_1 = .5\{1 - 2.7281/\sqrt{8\pi}\} = .12712$$

and

$$P_2 = .5\{.93845/\sqrt{8\pi}\} = .0936$$

and so

$$P = P_1 + P_2 = .2207$$

This means that this discriminant function could be expected to classify continuous Gaussian patterns with a mean and variance as given with approximately 22% misclassification, 9.3% from category two and 12.7% from category one.

4.5 Comparison of the Two Systems

The two systems used for training in the last two sections differ very much in their capabilities for finding the optimal solution. The mean-square-error classifier has a smooth convergence so that every change in the weight vector is an improvement. The convergence rate, however, is very slow. The reason for the poor convergence rate appears to be more the fault of the performance criterion than the search technique. For example, the optimal solution $f(X) = -4x_1 - x_2 + 5$ gives a mean-square-error of 38.45. The mean-square-error during training remains below this value after the third step of learning and yet the solution never reaches the optimal. The reason for this is that the mean-square-error performance criterion assumes the discriminant function to be a mapping. Thus, multiplying f(X) by an appropriate constant can produce a far greater change in mean-square-error than a change in

individual elements of the weight vector even though this multiplication by a constant does not change the solution. This performance criterion then places the greater importance on the absolute magnitudes of the weight vector elements and a far lesser importance on their relative magnitudes. However, for classification, absolute magnitudes are of no importance, since the solution is determined by the relative magnitudes together with the signs of the individual elements.

The mean-square-error training procedure requires slightly more than twenty seconds to process 100 samples. This means the complete training process took over thirty-three minutes of computer time to complete. On the other hand, the probabilistic-descent process only calculates a change in W when a sufficiently large error in f(X) occurs. It, therefore, takes less time to do the 100 samples, a little more than fifteen seconds. However, for this method the training procedure need only be repeated 50 times to obtain the solution given and so the total time required will only be about thirteen minutes. This time can be compared with one minute and thirty seconds, the time given by Patterson and Womack for their solution. Although the information given in the paper by Patterson and Womack is not very complete, their results show they have obtained a better solution in far less computing time than we have been able to. However, because the probability of error produced was low as explained in section 4.2, one possible explanation for the discrepancy might be that the sample set used by these authors was a poor representation of a Gaussian distribution.

By comparing Figure 4.1 with Figure 4.4, we find the major difference in the physical structure of the two systems to be the amount

of memory required. In the probabilistic-descent classifier, the only memory required is for the evaluation of the matrix E. This would make this scheme more practical for implementation.

The probabilistic-descent training process has one quality which is inferior to the mean-square-error training procedure and this is the fluctuation of the convergence, compared to the smooth convergence of the mean-square-error procedure. In the probabilistic-descent method, the solution moves toward the optimal as an average but each individual change in the weight vector is not guaranteed to improve the solution. This disadvantage is, however, compensated by the increased convergence rate. A summary of the results of the two training procedures is given in Table 4.2.

Table 4.2
SUMMARY OF SIMULATION RESULTS

TRAINING METHOD	DISCRIMINANT FUNCTION	ERRORS PRODUCED	100xPROBABILITY OF ERROR IN A CONTINUOUS DISTRIBUTION	TIME REQUIRED FOR SOLUTION
Bayes Optimal Solution	$-4x_1 - x_2 + 5$	15	13	-
Mean-Square-Error	$-.2253x_1^2 - .03314x_2^2 - .06276x_1x_2 + .2987x_1 + .09026x_2 + .8023$	36	36.68	33 minutes
Probabilistic-Descent	$-3.2896x_1^2 - .04201x_2^2 - .7481x_1x_2 + 1.8927x_1 - 1.1685x_2 + 5.5026$	16	22.07	13 minutes

CHAPTER 5 CONCLUSIONS

5.1 Summary

In this thesis, we have tried to present the reader with enough knowledge about the field of pattern classification to allow him to pursue the topic in the current literature. The major portions of chapters two and three consist of material collected from many different sources, most of which are listed in the bibliography. Some parts, such as section 2.5 and parts of section 3.2, are, however, original ideas which, it is hoped, will add to the ideas already available. The work of chapter four has been carried out to familiarize the author with an actual classification problem and to check the validity of an existing publication. It is hoped that more work can be done on this problem and that the results can be discussed with the authors of that publication.

5.2 Some General Comments

Certain aspects of pattern classification have been studied thoroughly while other aspects have received relatively little attention. For example, linear classifiers discussed in section 2.2 have been studied very thoroughly and many of the practical examples that have been produced are linear machines e.g. ADALINE[2]. On the theoretical side, Highlyman[8] has done a complete Doctoral Dissertation on linear decision functions. On the other hand, the study of layered machines has not progressed very far. One of the main reasons seems to be the problem of training.

In the area of training, new nonparametric training methods for linear machines seem to be appearing continually in the current

literature. These training methods can also be applied to any Φ machine since a Φ machine is linear with respect to its weight vector. New methods of parametric training are not as easy to produce. The methods already available, most of which can be found in the review of Nagy[14], are generally accepted as adequate for most parametric training problems.

5.3 Possible Areas for Further Research

There still are many basic problems to be solved before the full potential of learning machines can be realized. One such problem is the formulation of a transfer function for a linear machine. If this could be done, networks of these machines could be both analyzed and synthesized with much less difficulty and far more complex classifiers could be studied. Another problem which has not as yet been solved is the training of more complex machines such as layered machines. This problem could be very well worth studying because, as was shown in this thesis, any two classes which are separable by a polynomial function, no matter how complex, can be separated by a layered machine preceded by a Φ processor. In the field of parametric training, the basic training problems remaining include the introduction of new probability functions and methods for obtaining the parameters for them.

The future of this study of pattern recognition by adaptive pattern classifiers seems to lie in the applications of the study. As yet, the theoretical knowledge has far out-run the experimental successes. Other areas of study seem to be picking up the ideas already presented and using them. One of the best examples is the field of control systems where both on-line and off-line learning techniques are receiving an increased amount of attention[18]. This research is leading to control

systems known as self-organizing systems. Such systems are defined as systems which change their basic structure as a function of experience and/or environment[20]. Widrow[4] has even mentioned the possibility of self-repairing systems. It would take little imagination to realize the significance of such systems. In the future, as more of these applications are developed, the demand for research in the area of pattern classification will increase.

BIBLIOGRAPHY

1. Widrow, B.: "Generalization and Information Storage in Networks of Adaline 'Neurons'"; Yovits, Jacobi and Goldstein, "Self-organizing Systems - 1962"; Spartan Books, Washington, D.C.; pp. 435-461; 1962.

2. Widrow, B.: "An Adaptive ADALINE Neuron Using Chemical Memistors"; Stanford University ERL Tech. Report No. 1553-2; October 17, 1960.

3. Huber, William A.: "Learning Machine Techniques for Pattern Classification"; Proceedings of the National Electronics Conference; vol. 21; McCormick Place, Chicago, Illinois; pp. 517-522; October 26, 27 and 28, 1965.

4. Widrow, B., G.F. Groner, M.J.C. Hu, F.W. Smith, D.F. Specht and L.R. Talbert: "Practical Application for Adaptive Data-Processing Systems"; Review A; Brussels 6, Belgium; January, 1968.

5. Darling, Eugene M., and R. David Joseph: "Pattern Recognition from Satellite Altitudes"; IEEE Transactions on Systems Science and Cybernetics, vol. SSC-4, No. 1, pp. 38-47; March, 1968.

6. Feller, William: "An Introduction to Probability Theory and its Applications, vol. 1"; 2nd Edition; John Wiley and Sons Inc.; New York, N.Y.; p. 36; 1957.

7. Nilsson, N.J.: "Learning Machines: Foundations of Trainable Pattern-Classifying Systems"; McGraw-Hill Book Company; New York, 1965.

8. Highleyman, W.H.: "Linear Decision Functions, with Application to Pattern Recognition"; Proc. IRE, vol. 50, part 2 pp. 1501-1514, 1962.

9. Anderson, T.W.: "An Introduction to Multivariate Statistics"; John Wiley and Sons Inc., New York, N.Y.; 1958.

10. Kashyap, R.L. and Colin C. Blaydon: "Estimation of Probability Density and Distribution Functions"; IEEE Transactions on Information Theory, vol. IT-14 No. 4, pp. 549-556; July, 1968.

11. Cooper, P.W.: "Multivariate Extension of Univariate Distributions"; IEEE Transactions on Electronic Computers.

12. Ahmed, N.U.: "A Sequence of Probability Measures on the Euclidean n-space and its Limit"; AECD/EL/6; Dacca, Pakistan; May, 1967.

13. Rosenblatt, F.: "Principles of NEURODYNAMICS"; Spartan Books, Washington, D.C.; pp. 111-116; 1962.

14. Nagy, George: "Classification Algorithms in Pattern Recognition"; IEEE Transactions on Audio and Electroacoustics, vol. AU-16, No. 2; June, 1968.

15. Chien, Y.T.: "Simultaneous Detection and Estimation of Pattern Characteristics in Self-Learning Systems"; Presented to the 6th Annual Allerton Conference on Circuit and System Theory; Allerton House, Monticello, Illinois; October 2-4, 1968.

16. Patterson, J.D. and B.F. Womack: "An Adaptive Pattern Classification System"; IEEE Transactions on Systems Science and Cybernetics, vol. SSC-2, No. 1; pp. 62-67; August, 1966.

17. Amari, Shunichi: "A Theory of Adaptive Pattern Classifiers"; IEEE Transactions on Electronic Computers, vol. EC-16, No. 3; June, 1967.

18. Mendel, J.M. and J.J. Zapalac: "Realization of a Suboptimal Controller by Off-Line Training Techniques"; Preprints of JACC 67 papers; University of Pennsylvania; pp. 258-266; June 28-30, 1967.

19. Hamming, R.W.: "Numerical Methods for Scientists and Engineers"; McGraw-Hill, N.Y.; pp. 34 and 389; 1962.

20. Mendel, J.M. and J.J. Zapalac: "Advances in Control Systems"; vol. 6; 1968, Academic Press, New York and London; chapter 1.

APPENDIX A CALCULATION OF THE ERROR PROBABILITY FOR THE BAYE'S SOLUTION

Given a Gaussian probability distribution we get:

$$\int_{V_2=\{X:f(X)<0\}} p_1(X)dX = \frac{1}{2\pi|\Sigma|^{1/2}} \int_{V_2=\{X:f(X)<0\}} \exp\{-1/2(X-M)^t \Sigma^{-1}(X-M)\}dX \qquad (A.1)$$

With $M = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$ this becomes

$$\frac{1}{4\pi} \int_{x_2=-\infty}^{+\infty} e^{-x_2^2/8} \int_{x_1=g(x_2)}^{+\infty} e^{-x_1^2/2} dx_1 dx_2 \qquad (A.2)$$

where $g(x_2)$ results from solving the discriminant function at the decision boundary for x_1. The second integral in A.2 can be considered as 2 separate integrals.

$$\int_{x_1=g(x_2)}^{+\infty} e^{-x_1^2/2} dx_1 = \int_{o}^{\infty} e^{-x_1^2/2} dx_1 - \int_{x_1=0}^{g(x_2)} e^{-x_1^2/2} dx_1 \qquad (A.3)$$

The first integral on the right hand side of A.3 can be solved by standard methods (see ref. 6, p.164-166). This gives

$$\int_{o}^{\infty} e^{-x_1^2/2} dx_1 = \sqrt{\frac{\pi}{2}} \qquad (A.4)$$

For the second integral let $u = \frac{x_1}{\sqrt{2}}$; $du = \frac{dx_1}{\sqrt{2}}$

this gives

$$\int_{x_1=0}^{g(x_2)} e^{-x_1^2/2} dx_1 = \int_{u=0}^{u=\frac{g(x_2)}{\sqrt{2}}} \sqrt{2}\, e^{-u^2} du$$

$$= \sqrt{\frac{\pi}{2}}\, \mathrm{erf}\left[\frac{g(x_2)}{\sqrt{2}}\right] \qquad (A.5)$$

where we use the definition

$$\mathrm{erf}(X) = \frac{2}{\sqrt{\pi}} \int_{o}^{X} e^{-u^2}\, du$$

we therefore have the result

$$\int_{x_1=g(x_2)}^{+\infty} e^{-x_1^2/2}\, dx_1 = \sqrt{\frac{\pi}{2}} \left\{1 - \mathrm{erf}\left[\frac{g(x_2)}{\sqrt{2}}\right]\right\} \qquad (A.6)$$

Substituting this result into equation A.2 gives

$$\frac{1}{4\sqrt{2\pi}} \int_{x_2=-\infty}^{+\infty} e^{-x_2^2/8} \left[1 - \mathrm{erf}\left(\frac{g(x_2)}{\sqrt{2}}\right)\right] dx_2 \qquad (A.7)$$

The first term in A.7 can be determined by using the same method applied to the first term in A.3. This gives

$$\int_{x_2=-\infty}^{+\infty} e^{-x_2^2/8}\, dx_2 = \sqrt{8\pi} \qquad (A.8)$$

The second term in A.7 cannot be solved analytically and so a computer integration method is used. From equation 4.1 the function $g(x_2)$ can be determined.

$$g(x_2) = \frac{5 - x_2}{4} \qquad (A.9)$$

Substituting this into the second term in A.7 and using a computer integration gives

$$\int_{x_2=-\infty}^{+\infty} e^{-x_2^2/8}\, \mathrm{erf}\left[\frac{(5 - x_2)}{4\sqrt{2}}\right] dx_2 = 3.692 \qquad (A.10)$$

We therefore get

$$P(\mathrm{ERROR}) = \frac{1}{4\sqrt{2\pi}} \{\sqrt{8\pi} - 3.692\} = .1317 \qquad (A.11)$$

For the second integral it is most convenient to integrate $p_1(X)$ over the space for which it is correctly classified and subtract this from 1.

The decision surface for this case is a parabola symmetric about the x_1-axis. $\therefore$ the probability of error is

$$P(\text{ERROR}) = 1 - \int_{V_1=\{X:f(X)>0\}} p_1(X)\,dX = 1 - \frac{2}{4\pi} \int_{x_1=-\infty}^{a} e^{-x_1^2/2} \int_{x_2=0}^{g_2(x_1)} e^{-x_2^2/8}\, dx_2 dx_1 \qquad (A.12)$$

where $g_2(x_1)$ is the positive root of $x_2 = \{\frac{86.18 - 64x_1}{3}\}^{1/2}$

and "a" is the value of x_1 for which $x_2 = f(x_1) = 0$

i.e. $a = \frac{86.18}{64} = 1.347.$

Using a similar method to that used in A.5

$$\int_{x_2=0}^{g_2(x_1)} e^{-x_2^2/8}\, dx_2 = \int_{u=0}^{u=\frac{g_2(x_1)}{\sqrt{8}}} \sqrt{8}\, e^{-u^2}\, du = \sqrt{2\pi}\ \text{erf}\ \frac{g_2(x_1)}{\sqrt{8}} \qquad (A.13)$$

The remaining integral is solved by computer methods

$$\frac{\sqrt{2\pi}}{2\pi} \int_{x_1=-\infty}^{x_1=a=1.347} e^{-x_1^2/2}\ \text{erf}\ \{\frac{g_2(x_1)}{\sqrt{8}}\} = .8693 \qquad (A.14)$$

$$\therefore P(\text{ERROR}) = .1307$$

The third decision boundary is a hyperbola symmetric about both the x_1 and x_2 axes. We therefore need only integrate the distribution over one quadrant and multiply by 4 to get the complete integral.

$$P(\text{ERROR}) = \frac{4}{4\pi} \int_{x_1=-\infty}^{a} e^{-x_1^2/2} \int_{x_2=0}^{g_3(x_1)} e^{-x_2^2/8}\, dx_2 dx_1 \qquad (A.16)$$

The discriminant function for this case is

$$f(X) = \frac{22.18 - 15x_1^2 + 12x_2^2}{32}$$

so $$g_3(x_1) = \{\frac{15x_1^2 - 22.18}{12}\}^{1/2}$$

and $$a = \sqrt{\frac{22.18}{15}} = \pm\sqrt{1.48}$$

From (A.13) $$\int_{x_2=0}^{g_3(x_1)} e^{-x_2^2/8}\, dx_2 = \sqrt{2\pi}\ \text{erf}\ \{\frac{g_3(x_1)}{\sqrt{8}}\} \qquad (A.17)$$

Again, using computer methods gives

$$\sqrt{\frac{2}{\pi}} \int_{x_1=-\infty}^{a=-1.215} e^{-x_1^2/2}\ \text{erf}\ \{\frac{g_3(x_1)}{\sqrt{8}}\}\, dx_1 = .09886 \qquad (A.18)$$

and so P(ERROR) = .0989 (A.19)

The fourth decision boundary is an ellipse and so it will be much easier to integrate $p_1(X)$ over V_1 and subtract this value from one. Again quadrant symmetry allows us to integrate over one quadrant.

$$P(\text{ERROR}) = 1 - \frac{4}{4\pi} \int_{x_1=0}^{a} e^{-x_1^2/2} \int_{x_2=0}^{g_4(x_1)} e^{-x_2^2/8}\, dx_2 dx_1 \qquad (A.20)$$

In this case the discriminant function is

$$f(X) = \frac{44.35 - 12x_1^2 - 3x_2^2}{32}$$

and so $$g_4(x_1) = \{\frac{44.35 - 12x_1^2}{3}\}^{1/2}$$

and $$a = \sqrt{\frac{44.35}{12}} = \sqrt{3.696} = 1.922$$

Again from A.13 $$\int_{x_2=0}^{g_4(x_1)} e^{-x_2^2/8}\, dx_2 = \sqrt{2\pi}\ \text{erf}\ \{\frac{g_4(x_1)}{\sqrt{8}}\} \qquad (A.21)$$

From the computer

$$\sqrt{\frac{2}{\pi}} \int_{x_1=0}^{x_1=a=1.922} e^{-x_1^2/2} \operatorname{erf}\left\{\frac{g_4(x_1)}{\sqrt{8}}\right\} dx_1 = .84256 \qquad (A.22)$$

$$\text{so } P(\text{ERROR}) = 1 - .84256 = .15744 \qquad (A.23)$$

APPENDIX B CALCULATION OF THE GRADIENT OF THE MEAN-SQUARE-ERROR

We would like to solve for a value of s and λ which will produce a minimum value for $M(W + \lambda s)$. We have assumed a quadratic discriminant function.

$$f(W,X) = w_1x_1^2 + w_2x_2^2 + w_3x_1x_2 + w_4x_1 + w_5x_2 + w_6$$

so we can write

$$f(W + \lambda s,X) = (w_1 + \lambda s_1)x_1^2 + (w_2 + \lambda s_2)x_2^2 + (w_3 + \lambda s_3)x_1x_2$$

$$+ (w_4 + \lambda s_4)x_1 + (w_5 + \lambda s_5)x_2 + w_6 + \lambda s_6$$

$$= f(W,X) + f(\lambda s,) \qquad (B.1)$$

We can then expand the terms in $M(W + \lambda s)$ using the form given in (B.1)

$$\{f(W,X_i) + f(\lambda s,X_i) + C\}^2 = \{f(W,X_i) + C\}^2 + \{f(\lambda s,X_i)\}^2$$

$$+ 2f(W,X_i)\ f(\lambda s,X_i) + 2f(\lambda s,X_i)\ C \qquad (B.2)$$

for $i = \alpha$ or β and $C = -C(2/1)$ for α and $C = +C(1/2)$ for β.

$$M(W + \lambda s) = M(W) + \frac{q_1}{N_1}\sum_{\alpha=1}^{N_1}\{[f(\lambda s, X_2)]^2 + 2f(\lambda s,X_\alpha)[f(W,X_\alpha)$$

$$-C(2/1)]\} + \frac{q_2}{N_2}\sum_{\beta=1}^{N_2}\{[f(\lambda s,X_\beta)]^2$$

$$+ 2f(\lambda s,X_\beta)[f(W,X_\beta) + C(1/2)]\} \qquad (B.3)$$

We also can see from B.1

$$f(\lambda s,X) = \lambda f(s,X)$$

so

$$M(W + \lambda s) = M(W) + \lambda^2\frac{q_1}{N_1}\sum_{\alpha=1}^{N_1}[f(s,X_\alpha)]^2 + \frac{q_1}{N_1} \times 2\lambda\sum_{\alpha=1}^{N_1} f(s,X_\alpha)$$

$$[f(W,X_\alpha) - C(2/1)] + \lambda^2\frac{q_2}{N_2}\sum_{\beta=1}^{N_2}[f(s,X_\beta)]^2 +$$

$$+ \frac{q_2}{N_2} \times 2\lambda \sum_{\beta=1}^{N_2} f(s,X_\beta) \left[f(W,X_\beta) + C(1/2) \right] \qquad (B.4)$$

Let

$$G_\alpha^{(1)}(s) = \frac{q_1}{N_1} \sum_{\alpha=1}^{N_1} \left[f(s,X_\alpha) \right]^2 \; ; \; G_\beta^{(1)}(s) = \frac{q_2}{N_2} \sum_{\beta=1}^{N_2} \left[f(s,X_\beta) \right]^2$$

$$G_\alpha^{(2)}(s) = \frac{q_1}{N_1} \sum_{\alpha=1}^{N_1} f(s,X_\alpha) \left[f(W,X_\alpha) - C(2/1) \right]$$

$$G_\beta^{(2)}(s) = \frac{q_2}{N_2} \sum_{\beta=1}^{N_2} f(s,X_\beta) \left[f(W,X_\beta) + C(1/2) \right]$$

So

$$M(W + \lambda s) + M(W) + \lambda^2 \left[G_\alpha^{(1)}(s) + G_\beta^{(1)}(s) \right] + 2\lambda \left[G_\alpha^{(2)}(s) + \right.$$
$$\left. + G_\beta^{(2)}(s) \right] \qquad (B.5)$$

$$\frac{dM(W + \lambda s)}{d\lambda} = 2\lambda \left[G_\alpha^{(1)}(s) + G_\beta^{(1)}(s) \right] + 2 \left[G_\alpha^{(2)}(s) + \right.$$
$$\left. + G_\beta^{(2)}(s) \right] \qquad (B.6)$$

Setting (B.6) to zero gives a minimum value to $M(W + \lambda s)$. So the value of λ which minimizes $M(W + \lambda s)$ is given by

$$\lambda = - \frac{G_\alpha^{(2)}(s) + G_\beta^{(2)}(s)}{G_\alpha^{(1)}(s) + G_\beta^{(1)}(s)} = - \frac{G^{(2)}(s)}{G^{(1)}(s)} \qquad (B.7)$$

where

$$G^{(i)}(s) = G_\alpha^{(i)}(s) + G_\beta^{(i)}(s) \quad \text{for } i = 1,2$$

This value gives a true minimum because $G^{(1)}(s) \geq 0$ which means $M(W + \lambda s)$ is a convex function of λ.

$\left. \frac{dM}{d\lambda} \right|_{\lambda=0}$ is the gradient of $M(W)$ at W. So to find s such that the

gradient is maximum we must take

$$\max \left.\frac{dM}{d\lambda}\right|_{\lambda=0} = \max 2G^{(2)}(s)$$

$$G^{(2)}(s) = \frac{q_1}{N_1} \sum_{\alpha=1}^{N_1} f(s,X_\alpha) \left[f(W,X_\alpha) - C(2/1)\right] +$$

$$\frac{q_2}{N_2} \sum_{\beta=1}^{N_2} f(s,X_\beta) \left[f(W,X_\beta) + C(1/2)\right]$$

We can define

$$f(s,X) = (s \cdot F(X))$$

where F(X) is the Φ function of the pattern vector.

$$F(X) = (X_1^2,\ X_2^2,\ X_1X_2,\ X_1,\ X_2,\ 1)$$

We can further define $(s \cdot F(X)) \times (W \cdot F(X)) = (As \cdot W)$

by defining A as the outerproduct of F(X) with itself.

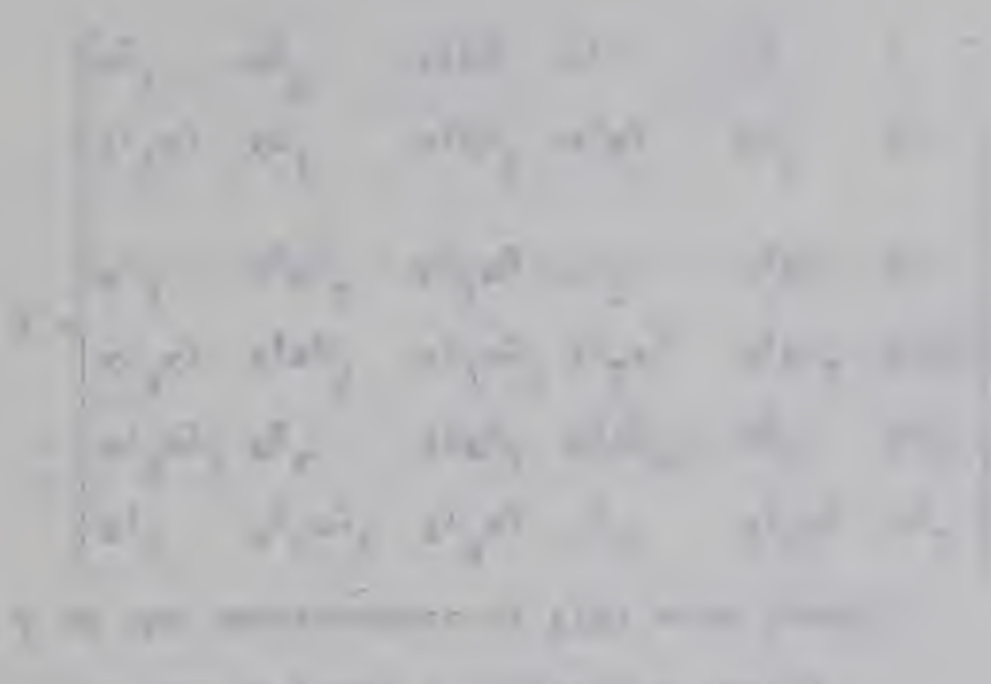
$$A = \begin{bmatrix} x_1^4 & x_1^2x_2^2 & x_1^3x_2 & x_1^3 & x_1^2x_2 & x_1^2 \\ x_1^2x_2^2 & x_2^4 & x_1x_2^3 & x_1x_2^2 & x_2^3 & x_2^2 \\ x_1^3x_2 & x_1x_2^3 & x_1^2x_2^2 & x_1^2x_2 & x_1x_2^2 & x_1x_2 \\ x_1^3 & x_1x_2^2 & x_1^2x_2 & x_1^2 & x_1x_2 & x_1 \\ x_1^2x_2 & x_2^3 & x_1x_2^2 & x_1x_2 & x_2^2 & x_2 \\ x_1^2 & x_2^2 & x_1x_2 & x_1 & x_2 & 1 \end{bmatrix}$$

So

$$G^{(2)}(s) = \frac{q_1}{N_1} \sum_{\alpha=1}^{N_1} \left[(A(X_\alpha)s \cdot W) - (s \cdot F(X_\alpha)) \times C(2/1)\right]$$

$$+ \frac{q_2}{N_2} \sum_{\beta=1}^{N_2} \left[(A(X_\beta)s \cdot W) + (s \cdot F(X_\beta)) \times C(1/2)\right] \qquad (B.8)$$

$$(A(X)s \cdot W) = (s \cdot A^t(X)W)$$



so

$$G^{(2)}(s) = \frac{q_1}{N_1} \sum_{\alpha=1}^{N_1} (s \cdot \{A^t(X_\alpha)W - F(X_\alpha) \times C(2/1)\})$$

$$+ \frac{q_2}{N_2} \sum_{\beta=1}^{N_2} (s \cdot \{A^t(X_\beta)W + F(X_\beta) \times C(1/2)\})$$

$$= (s \cdot \{ \frac{q_1}{N_1} \sum_{\alpha=1}^{N_1} A^t(X_\alpha)\ W - \frac{q_1}{N_1} \sum_{\alpha=1}^{N_1} F(X_\alpha) \times C(2/1)$$

$$+ \frac{q_2}{N_2} \sum_{\beta=1}^{N_2} A^t(X_\beta)\ W + \frac{q_2}{N_2} \sum_{\beta=1}^{N_2} F(X_\beta) \times C(1/2)\}) \qquad (B.9)$$

Now, by Schwartz inequality, maximum $G^{(2)}(s)$ will be given by

$$s = \frac{q_1}{N_1} \left[\left(\sum_{\alpha=1}^{N_1} A^t(X_\alpha) \right) W - \sum_{\alpha=1}^{N_1} F(X_\alpha) \times C(2/1) \right]$$

$$+ \frac{q_2}{N_2} \left[\left(\sum_{\beta=1}^{N_2} A^t(X_\beta) \right) W + \sum_{\beta=1}^{N_2} F(X_\beta) \times C(1/2) \right] \qquad (B.10)$$

APPENDIX C CALCULATION OF THE ERROR PROBABILITY FOR THE MEAN-SQUARE-ERROR SOLUTION

$$P = q_1\{1 - \frac{1}{4\pi}\int_{x_1=+a_1}^{x_1=a_2} e^{-\frac{1}{2}x_1^2}\int_{x_2=f_1(x_1)}^{f_2(x_1)} e^{-x_2^2/8}\,dx_2dx_1\} +$$

$$+ q_2\{1 - \frac{1}{4\pi}\int_{x_1=a_1}^{a_2} e^{-(x_1-2)^2/2}\int_{x_2=f_1(x_1)}^{f_2(x_1)} e^{-(x_2-2)^2/8}dx_2dx_1 \qquad (C.1)$$

The decision boundary is defined by

$$-.2253x_1^2 - .03314x_2^2 - .06276x_1x_2 + .2987x_1 + .09026x_2 + .8023 = 0 \qquad (C.2)$$

Solving for x_2 gives

$$x_2 = .06276x_1 - .09026 \pm (h(x_1))^{\frac{1}{2}} \;/\; 2(-.03314) \qquad (C.3)$$

where

$$h(x_1) = (.06276x_1 - .09026)^2 + 4(.03314)(.8023 + .2987x_1 - .2253x_1^2)$$

Taking the negative square root will give $f_1(x_1)$ and taking the positive square root will give $f_2(x_1)$.

To find a_1 and a_2 we must solve the square root term for the value of x_1 which gives 0.

$$0 = (.06276x_1 - .09026)^2 + 4(.03314)(.8023 + .2987x_1 - .2253x_1^2)$$

$$0 = -.02592695x_1^2 + .02826623x_1 + .11339876 \qquad (C.4)$$

which yields $x_1 = -1.626$ and 2.716 so $a_1 = -1.626$ and $a_2 = 2.716$

To solve the integral

$$\int_{x_2=f_1(x_1)}^{f_2(x_1)} e^{-x_2^2/8}\,dx_2$$

we take

$$\int_{x_2=f_1(x_1)}^{f_2(x_1)} e^{-x_2^2/8}\, dx_2 = \int_{x_2=f_1(x_1)}^{0} e^{-x_2^2/8}\, dx_2 + \int_{0}^{f_2(x_1)} e^{-x_2^2/8} dx_2 \qquad (C.5)$$

let $u = \frac{x_2}{\sqrt{8}}$; $dx_2 = \sqrt{8}\, du$

$$\int_{x_2=f_1(x_1)}^{f_2(x_1)} e^{-x_2^2/8}\, dx_2 = -\int_{u=0}^{f_1(x_1)/\sqrt{8}} e^{-u^2} \sqrt{8}\, du + \int_{u=0}^{f_2(x_1)/\sqrt{8}} e^{-u^2}\, \sqrt{8}\, du$$

$$= -\frac{\sqrt{8\pi}}{2} \operatorname{erf}\left[\frac{f_1(x_1)}{\sqrt{8}}\right] + \frac{\sqrt{8\pi}}{2} \operatorname{erf}\left[\frac{f_2(x_1}{\sqrt{8}}\right] \qquad (C.6)$$

So

$$P_1 = q_1\{1 - \frac{\sqrt{8\pi}}{4\pi\times 2} \int_{a_1}^{a_2} e^{-\frac{1}{2}x_1^2} \left[\operatorname{erf} \frac{f_2(x_1)}{\sqrt{8}} - \operatorname{erf} \frac{f_1(x_1)}{\sqrt{8}}\right] dx_1\} \qquad (C.7)$$

Doing the integration by computer gives

$$P_1 = q_1\{1 - \frac{\sqrt{8\pi}}{8\pi}(4.2686)\} = .5\{1 - .8515\} = .0743 \qquad (C.8)$$

In a similar manner P_2 can be found.

Let

$$u = \frac{x_2 - 2}{\sqrt{8}} \; ; \; dx_2 = \sqrt{8}\, du$$

$$\int_{x_2=f_1(x_1)}^{f_2(x_1)} e^{-(x_2-2)^2/8}\, dx_2 = -\int_{u=0}^{f_1(x_1)-2/\sqrt{8}} \sqrt{8}\; e^{-u^2} du +$$

$$+ \int_0^{f_2(x_1)-2} \sqrt{8}\, e^{-u^2}\, du$$

$$= -\sqrt{8}\,\frac{\sqrt{\pi}}{2}\,\mathrm{erf}\,\frac{f_1(x_1)-2}{\sqrt{8}} + \sqrt{8}\,\frac{\sqrt{\pi}}{2}\,\mathrm{erf}\,\frac{f_2(x_1)-2}{\sqrt{8}} \qquad (C.9)$$

So

$$P_2 = q_2\{\frac{\sqrt{8\pi}}{8\pi} \int_{x_1=a_1}^{a_2} e^{-(x_1-2)^2/2}\,(\mathrm{erf}\left[\frac{f_2(x_1)-2}{\sqrt{8}}\right] -$$

$$\mathrm{erf}\left[\frac{f_1(x_1)-2}{\sqrt{8}}\right])\, dx_1\} \qquad (C.10)$$

Again the integration had to be done by computer methods giving

$$P_2 = q_2\{\frac{\sqrt{8\pi}}{8\pi}\,(2.9332)\} = .5(.5851) = .2925$$